Semi-Supervised Topic Models Applied to Mathematical Document Classication

Evans, Ieuan

Award date:
2017

Awarding institution:
University of Bath

Semi-Supervised Topic Models Applied to Mathematical Document Classification

submitted by

Ieuan Evans

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Science

March 2017

Summary

Our objective is to build a mathematical document classifier: a machine which for a given mathematical document $\mathbf{x}$, determines the mathematical subject area $\mathbf{c}$. In particular, we wish to construct the function f such that $f(\mathbf{x}, \mathbf{\Theta}) = \mathbf{c}$ where f requires the possibly unknown parameters $\mathbf{\Theta}$ which may be estimated using an existing corpus of labelled documents. The novelty here is that our proposed classifiers will observe a mathematical document over dual vocabularies. In particular, as a collection of both words and mathematical symbols.

In this thesis, we predominantly review the claims made in [1]: mathematical document classification is possible via symbol frequency analysis. In particular, we investigate whether this claim is justified: [1] contains no experimental evidence which supports this. Furthermore, we extend this research further and investigate whether the inclusion of mathematical notational information improves classification accuracy over the existing single vocabulary approaches. To do so, we review a selection of machine learning methods for document classification and refine and extend these models to incorporate mathematical notational information and investigate whether these models yield higher classification performance over existing word only versions.

In this research, we develop the novel mathematical document models "Dual Latent Dirichlet Allocation" and "Dual Pachinko Allocation" which are extensions to the existing topic models "Latent Dirichlet Allocation" and "Pachinko Allocation" respectively. Our proposed models observe mathematical documents over two separate vocabularies (words and mathematical symbols). Furthermore, we present Online Variational Bayes for Pachinko Allocation and our proposed models to allow for fast parameter estimation over a single pass of the data.

We perform systematic analysis on these models, and we verify the claims made in [1], and furthermore, we observe that the inclusion of symbol data via Dual Pachinko Allocation only yields in an increase of classification performance over the single vocabulary variants and the prior art in this field.

Acknowledgments

I would like to thank my supervisors Prof. James H. Davenport and Prof. Peter M. Hall for their continued support and helping me to keep my mind on my research when needed. I would also like to thank my friends, family and colleagues for helping me to take my mind off my research when needed.

Contents

Chapter 1

Introduction

In this thesis, we investigate automatic mathematical document classification: the technology to detect the subject area of new mathematical documents via machine learning techniques. The novelty here is that current approaches to document classification assume only a single vocabulary, unlike mathematical documents which, by nature, span dual vocabularies (natural language and mathematical notation).

The claim to be tested is that accurate mathematical document classification requires the inclusion of *both* the textual content and the mathematical notational content. To do so, we refine and extend existing document classification techniques which operate over single vocabularies and introduce novel document models which operate on dual vocabularies. Most notably, we introduce *Dual Pachinko Allocation* in Chapter 7.

1.1 Introduction to the Problem

The digital era has dramatically changed the ways that academics search, produce, publish and disseminate their scientific work [2]. For example, there are papers written in LaTeX and rendered to PDF files, which are in turn published to digital libraries such the arXiv; articles as web-pages on websites such as Wikipedia; or collections of digital scans of existing paper copies such as Google Books.

Vast amounts of mathematical documents are collected into online libraries

each month[1], and it is becoming increasingly important that this data be organised sufficiently well. Most importantly, these articles must be appropriately labelled and tagged so that they are easily locatable by prospective readers.

A consequence of such a volume of articles and documents being collected into these databases every month is that it is becoming harder and harder to enforce good quality labelling. For example, many documents remain either completely unlabelled, insufficiently labelled (e.g. only labelled with one of the several appropriate categories) or simply incorrectly labelled (for example, a lazy author may label a document as "General" so as not to leave the label empty). An automated process to relieve the workload of manually checking the accuracy and relevance of these labels is becoming more and more appealing since this is already a massively time-consuming process and will only get worse if the rates of submission continue to increase.

Intuitively, it seems that we as humans (with some prior mathematical knowledge) can identify the area of mathematics of an article by observing the mathematical notation used. For example, if we see the formula "$E = mc^2$" we may assume the article is about, or at least partly about, physics. In fact, [1] demonstrates that for a given mathematical subject area, the ordered set of the top six most frequently used symbols are unique to this field which suggests that there are possible grounds for a document classifier.

In the context of mathematical document classification, the inclusion of mathematical notation may introduce some potential difficulties since mathematics is an inherently complicated language compared to natural language. In particular, mathematics inherits the notions of homographs (words with the same spelling but different meanings) and synonyms (words with different spellings but with the same meaning). Mathematical homographs are symbols which have different meanings. For example, the mathematical symbol "$\leq$" is used to mean both "less than" and "is a subgroup of". Mathematical synonyms are when different symbols may have the same meaning. For example, the mathematical symbols "$\times$" and "$\cdot$" are both used to mean "multiply". These properties arise from the design of mathematics. In particular, when writing mathematics, the author may define and redefine notation as they wish. In natural language, however, it is uncommon for an author to redefine the meanings of words in a piece of text

[1] arXiv monthly submission rates: `https://arxiv.org/stats/monthly_submissions`

while still maintaining readability. We describe mathematical notation in more detail in Section 2.1.

The inspiration for handling symbols like words stems from the use of document classification techniques in many other classification settings, for example, photograph classification may be achieved by modelling photographs as collections of "visual" words which correspond to discrete visual features [3]. In this research, we propose novel mathematical document models which operate over dual vocabularies where we treat symbols like words, but from a separate vocabulary. In particular, we introduce Dual Latent Dirichlet Allocation (Chapter 5) which discovers correlations between words and symbols, and Dual Pachinko Allocation (Chapter 7) which captures correlations between words and correlations between symbols separately.

Since these models operate over collections of discrete data and not specifically words and symbols, any development to these existing document classification methods could be incorporated into other applications, specifically, scenarios where observations comprise of discrete data from two separable feature spaces (i.e. with features mapping to words and symbols). That is to say that the technology behind mathematical document classification is not problem domain specific.

1.2 Research Goals

In this section, we summarise the three main directions of this research.

First Direction of Research

Generative latent topic models such as the Latent Dirichlet Allocation model [4] and the Pachinko Allocation model [5] are popular in the field of machine learning but have not yet been explored in the context of *mathematical* document classification. We wish to investigate the classification performance of these topic models and compare to the prior art of mathematical document classification.

Second Direction of Research

We have seen that there are grounds for mathematical document classification via symbol frequency analysis:

- We aim to extend the Latent Dirichlet Allocation model to model mathematical documents comprising of both words and mathematical symbols as a random mixture of latent topics, namely *Dual Latent Dirichlet Allocation*.
- Similarly, via Pachinko Allocation we aim to extend Dual Latent Dirichlet Allocation to allow the modelling of the documents over mixtures of *disjoint* word topics and symbol topics, which we name *Dual Pachinko Allocation*.
- By investigating the classification performance of the above models, we wish to determine whether the inclusion of mathematical symbol data increases classification accuracy over the corresponding single vocabulary approaches.

Third Direction of Research

Finally, parameter estimation for Latent Dirichlet Allocation can be done efficiently and accurately by using an adapted Variational Bayes method (Online Variational Bayes) which requires only a single pass of the data [6]. We wish to develop an equivalent formulation of the Variational Bayes algorithms for Pachinko Allocation and our proposed models and further extend these algorithms to Online Variational Bayes.

1.3 Vocabulary and Notation

In this section, we outline common vocabulary and notation used in this thesis. The following list outlines the vocabulary used when discussing data and documents.

- A *word* is the standard unit of discrete textual data which index into a given vocabulary of words.
- A *symbol* is the standard unit of discrete mathematical notational data which index into a given vocabulary of mathematics. A symbol token does

not necessarily have to consist of one character: in this thesis, we consider all mathematical identifiers and operators as symbols which include compound symbols such as "$\leqq$" and function identifiers such as "sin". This notion is in line with the MathML *mathematical identifier* tokens described in Appendix 2.2.

- A *document* is a collection of words, a *mathematical document* is a collection of words *and* symbols. These are unordered collections; they are simply lists of the words/symbols which appear in the documents in no particular order.

- A *class* is a discrete label indexed from a set of given subject areas that may tag a document. For example, an author may tag a document with the classes corresponding to the labels according to the Mathematics Subject Classification scheme which we describe in Appendix A.2.

The following outlines the vocabulary used when discussing latent topic models.

- A word/symbol *topic* is a probability distribution of word/symbol instances over their respective vocabularies.

- A word/symbol *topic index* is a discrete index belonging to a word/symbol which indexes a set of word/symbol topics.

- A *topic mixture* is a vector of proportions of topic indices residing in a document. Alternatively, a topic mixture may also be considered as a probability distribution of word/symbol topic indices.

Table 1.1 outlines the mathematical notation used in this thesis.

	Probability Distributions
$\text{Dir}(\boldsymbol{\alpha})$	Dirichlet distribution with parameter $\boldsymbol{\alpha}$
$\text{Cat}(\boldsymbol{\theta})$	Categorical distribution with parameter $\boldsymbol{\theta}$
	Model Dimensions
V, V^s	Number of words/symbols in a vocabulary
N, N^s	Number of words/symbols in a document
K, K^s	Number of latent word/symbol topics of a corpus
S	Number of latent super-topics of a corpus
D	Number of documents in a corpus
	Data
$\mathbf{w}, \mathbf{s}, \mathbf{c}$	Word/symbol/class index, indicator vector
w, s, c	Word/symbol/class index, integer label
	Model Parameters
$\boldsymbol{\alpha}, \boldsymbol{\alpha}^s$	Dirichlet parameter(s) on document word/symbol topic mixtures
$\boldsymbol{\alpha}^r$	Dirichlet parameter on document super-topic mixtures
$\boldsymbol{\eta}, \boldsymbol{\eta}^s$	Dirichlet parameter on document word/symbol topics
$\boldsymbol{\Theta}$	The set of model parameters.
	Latent Variables
$\mathbf{z}, \mathbf{z}^s$	Word/symbol (super) topic indices, indicator vector
z, z^s	Word/symbol (super) topic indices, integer
$\mathbf{z}', \mathbf{z}'^s$	Word/symbol topic indices, indicator vector
z', z'^s	Word/symbol topic indices, integer
$\boldsymbol{\theta}, \boldsymbol{\theta}^s$	(Per super-topic) word/symbol topic mixtures
$\boldsymbol{\beta}, \boldsymbol{\beta}^s$	Per topic word/symbol mixtures
$\mathcal{H}$	The set of latent (hidden) variables
	Variational Parameters
$\boldsymbol{\gamma}, \boldsymbol{\gamma}^s$	Dirichlet parameter on word/symbol topic mixtures
$\boldsymbol{\gamma}^r$	Dirichlet parameter on super-topic mixtures
$\boldsymbol{\lambda}, \boldsymbol{\lambda}^s$	Dirichlet parameter on word/symbol topics
$\boldsymbol{\phi}, \boldsymbol{\phi}^s$	Multinomial parameter on word/symbol super-topic indices
$\boldsymbol{\phi}', \boldsymbol{\phi}'^s$	Multinomial parameter on word/symbol topic indices
$\mathcal{F}$	The set of free variational parameters

Table 1.1: Table of Notation

Chapter 2

Background

The main idea that underpins this thesis is that the accurate classification of mathematical documents by content requires both word and symbol data. In this chapter, we provide evidence to support the fact that we are indeed addressing a literature gap of importance.

We begin by developing an anecdotal argument in support of our intuition. We continue, by describing the literature that also addresses our problem, showing that others advance to same intuition and provide at least some empirical support for it; however, these approaches are vulnerable to some criticisms. We advocate a more principled approach: we outline document models with strong Bayesian foundations which assume strong statistical structures of the documents. These approaches, however, do not address our problem directly; they assume that classification over a single vocabulary is sufficient, whereas we assert the need for a dual vocabulary.

To summarise, the prior art either addresses our problem directly but in an unprincipled way using various ad hoc methods, or addresses classification in a principled way but assumes a weaker statistical structure than we assert. Our contribution is to investigate the classification of mathematical documents by content, using both words and symbols, in a principled way.

2.1 The Case for Using Notation

Mathematical notation is essential for communicating complex mathematical concepts and over many years, notational conventions have developed to ensure bet-

ter clarity of the mathematics presented [7]. Notational conventions vary from subject area to subject area. For example, in group theory the symbol "$\leq$" is more likely to mean "is a subgroup of" as opposed to the more familiar "is less than" relation. Similarly, capital letters G, H are more likely to denote groups where lower case letters i and n are more likely to indicate numbers which make snippets like $H \leq G$ and $i \leq n$ easier to interpret. In fact, we have seen the ensuing snippet in some group theory lecture notes:

Theorem:
"Suppose $H_i \leq G$ for $i \leq n$, then ..."

This statement is particularly interesting; in the lecturer's effort to save time and/or chalk, they can omit various details and still communicate the ideas. The lecturer does not explicitly declare what H, G, i and n are, and overloads the symbol "$\leq$" in a single statement, yet the student can still determine (perhaps with a little bit of effort) that the lecturer is likely trying to communicate the following:

Theorem:
"Let G be a group and suppose H_i are subgroups of G for integers $i = 1, 2, \ldots, n$ for some positive integer n, then ..."

Most authors write mathematics with these conventions in mind, yet they may wish to circumvent such conventions and possibly redefine or introduce new notation. In some cases, authors use these new notations without explanation which unfortunately means that there are some truly difficult pieces of mathematics to read [8].

To summarise, mathematics can be ambiguous and confusing. To address this, authors usually adhere to notational conventions, and in particular, the subject area strongly influences the notation used.

2.2 Types of Digital Mathematics

Here we list common examples of digitised mathematics.

PDF from LaTeX

The most common format for mathematical documents are rendered from LaTeX source code. LaTeX[1] is a document preparation system for high-quality typesetting, most often used for technical and scientific documents. As for mathematics, the author can provide the structure of any mathematics without having to worry about the presentation and formatting. For example, to produce the following piece of mathematics

$$E = \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}}$$

the author may type the following:

```
\begin{equation*}
    E=\frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}}
\end{equation*}
```

Notice that the author has provided no information about the positioning or font sizes of the various terms and that the fractions and square root operator have been rendered appropriately.

Most documents which are written in LaTeX are rendered to PDF, a file format which encapsulates all the information needed to display the document such as fonts and graphics. In most cases, when rendering LaTeX documents, the structure of the mathematics is lost, for example, in a PDF file, the above equation may be represented as an arbitrary set of characters with corresponding geometric locations. Such representations do not necessarily encode things like which superscript "2" belongs to which symbol which the author has declared in the above LaTeX snippet. It is worth noting here that there is progress in encoding this structural information in a PDF from LaTeX file [9] but this tends not to be done in practice since this requires significant amounts of effort (i.e. heavily modified LaTeX code and a modified version of pdfLaTeX).

MathML

MathML [10] is a application of XML designed to encode mathematics. In particular, MathML can be used to encode the *presentation* layer or the *content*

[1] `http://latex-project.org`

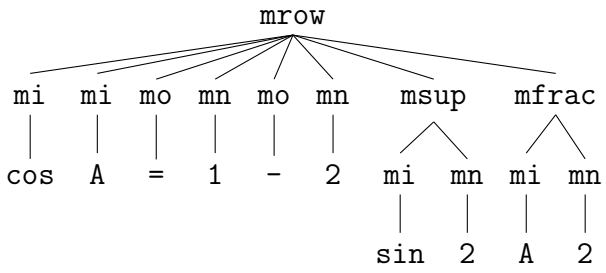

Figure 2-1: Graph of the MathML presentation markup of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$

layer of mathematics. The presentation layer only encodes how the mathematics is to be displayed, e.g. typesetting instructions. The content layer encodes the semantic meaning of the mathematics, i.e. explicitly and unambiguously describing the mathematical meaning. For example, the snippet of mathematics "$(0, 1)$" could be represented as a description of the presentation of the mathematics (the presentation layer), for example "*the Unicode symbols: 'left parenthesis', 'zero', 'comma', 'one', 'right parenthesis'*" or a description of the mathematics itself (the content layer), for example "*The open interval from zero to one*". Note that presentation and content cannot be related to a one to one mapping but instead a many-to-many mapping; mathematical notation (i.e. presentation) can be ambiguous, and that an author may present the same piece of semantics can in many ways. An example of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$ expressed in MathML presentation markup is given Appendix E.1.

Since XML is inherently a tree structure, we can visualise MathML script as a graph. Figure 2-1 shows the above expression in MathML presentation markup as a graph. The leaves of the graph are `<mi>`, `<mo>` and `<mn>` nodes which correspond to mathematical identifiers, operators and numbers respectively.

We remind ourselves that the above representation is *Presentation* MathML. In particular, we see that there is no information encoded to say the superscript in "$\sin^2$" corresponds to exponentiation (and not function composition), or more interestingly the concatenation "$2\sin$" is a multiplication but the concatenation "$\cos A$" is a function application. An example of the MathML content markup of the same expression is given in Appendix E.2 and Figure 2-2 shows the relevant graph.

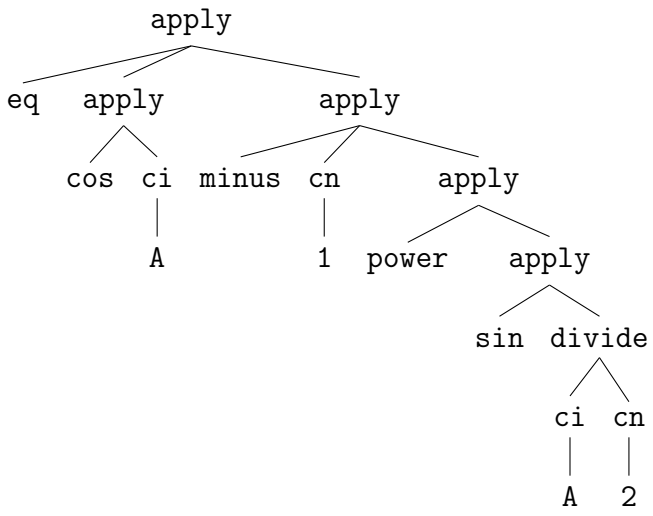

Figure 2-2: Graph of the MathML content markup of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$

Here the MathML content markup explicitly states the interaction between the various sub-expressions using the `<apply>` tag. Also notice that it avoids using the layout tags `<msup>` and `<mfrac>` and uses the explicit content operators `<power/>` and `<divide/>` instead. We note that some of these interactions could be explicitly stated in MathML presentation using "invisible" mathematical operators such as `<mo>⁢</mo>` and `<mo>&FunctionApply;</mo>`.

Meta-data

Meta-data refers to any additional information included with an article which describes any useful or relevant information about the article itself, which allows the efficient location of articles. For example, information about the author or publisher, and research areas covered by the article. The state of scientific publications relies heavily on the ability to search and retrieve relevant articles [2]. For example, according to [11], the most-cited articles are those of easiest access, in particular, those available online with complete meta-data.

2.3 Approaches to Mathematical Document Classification

In this section, we review and take inspiration from the current approaches to mathematical document classification. In particular, the work in [1] which investigates the use of symbol frequency analysis, and the work in [12] and [13] which explore general document classification techniques.

2.3.1 Symbol Frequency Analysis

The intuition in Section 2.1 suggests that the choice of mathematical symbols contain some extra information about the mathematical content and is not just an identifier. The observation that the symbols used depend on the subject area agrees with the assertion made in [1] which we now review.

The claim in this paper is that for each mathematical subject area, the ordered set of most commonly used symbols are unique, and furthermore, that mathematical document classification is possible via symbol frequency analysis. We identify a gap in the research here: although the author sufficiently justifies the claim with experimental evidence. There are no practical experiments which support the conclusion that document classification is possible via symbol frequency analysis. This observation is our motivation to investigate the use of mathematical notation in mathematical document classification.

2.3.2 Single Vocabulary Approaches

In this section we review the contributions mathematical document classification in [12] and [13]. We identify some concerns with the techniques used [12] which yields biased results. Furthermore, the authors in [13] highlight the same concerns and then present a solution to a realistic and unbiased problem setting. Briefly, the main difference between these two papers is that the work in [12] attempts to classify documents labelled with exactly one subject area, whilst the work in [13] endeavours to classify all subject areas of multi-labelled documents.

We first review the work in [12]. In this paper, the authors explore various combinations machine learning methods applied to mathematical document classification. The authors publish exciting results which boast extremely high

classification performance. We notice that there is a bias in the results here due to the nature of the problem setting. In particular, the data is aggressively pre-processed both at the document level and at the corpus level, and furthermore, articles are only considered if they are labelled with exactly one of the twenty most popular subject areas. As a result, the classifiers examined are trained and tested only on documents which are potentially the easiest to classify. Thus, this approach is unsuitable for real world applications.

We now review the work in [13]. In this paper, the authors address a more realistic problem setting: where documents can belong to many subject areas, and attempt to classify mathematical documents by observing their abstracts. Furthermore, the authors briefly describe a technique used to prevent over-fitting of the classifiers via the construction of balanced training and testing partitions. We describe this technique in detail in Section 3.5.

Finally, the authors briefly examine what effect the inclusion of the mathematical notation has on classification performance and demonstrate the possibility of performance gain via some empirical experiments. However, similar to [12], these experiments are performed on a carefully selected subset of the data. The authors discard all abstracts containing only trivial formulae or no formulae at all and thus this method is not directly applicable to real world applications.

2.4 Approaches to General Document Classification

In this section, we discuss the current approaches general document classification. The literature which we outline in Section 2.3 fundamentally base their work on discriminative models (in particular, SVM classification) which operate on large feature sets. Furthermore, these are fully supervised models; documents with missing labels are not included in the training corpora. These traditional discriminative classification methods used make little or no attempt to reveal the probabilistic structure and correlation within both input and output spaces [14].

There is an increasing interest in generative methods since these can exploit data with missing labels in addition to the labelled data [15]. We now review two popular generative document models. Firstly, we review Latent Dirichlet Allo-

cation, a document model based on the Dirichlet probability distribution, which assumes that the content of a document can be described as a mixture of latent (unobserved) topics. Secondly, we review Pachinko Allocation, a generalisation of Latent Dirichlet Allocation which imposes a stronger hierarchical statistical framework.

2.4.1 Latent Dirichlet Allocation

In this section, we briefly outline the Latent Dirichlet Allocation (LDA) model presented in [4] which we describe in detail in Chapter 4.

In the context of document modelling, Latent Dirichlet Allocation assumes that each document can be represented by a random mixture of latent topics, where these topics are each characterised by a distribution of words.

Topics can be thought of as anonymous bins of words. For example, one bin may contain the words "finance" and "interest"; another may contain the words "government" and "election"; and a third containing "health" and "hospital". A document may contain words from all bins, but may be weighted towards some bins more than others. These topics are not to be confused with subject areas; the topics of a document correspond to these anonymous distributions of words, the subject areas of a document correspond to observable labels which describe the content.

LDA can be used as an efficient filtering algorithm for feature selection without a decrease in classification performance. In particular, [16] shows that an SVM classifier trained on these topic mixtures yields an increase in classification performance over an SVM classifier trained on the word features directly. The feature selection of LDA is unsupervised; the parameter estimation step does not require labelled documents. In particular, this means any documents in the training corpus with missing labels can still be used to obtain the optimal topics and then discarded when performing supervised training on the labelled topic mixtures.

Inference in [4] is achieved via Batch Variational Bayes [17] which can be memory intensive, in particular, the algorithm requires many passes of the training corpus. In contrast, the Online Variational Bayes inference procedure for LDA outlined in [6] requires only a single pass of the training corpus.

2.4.2 Four-Level Pachinko Allocation

We now briefly outline the Pachinko Allocation model presented in [5], in particular, Four-Level Pachinko Allocation which we describe in detail in Chapter 6.

Pachinko Allocation provides a hierarchical generalisation of the Latent Dirichlet Allocation model. In particular, the Four-Level Pachinko Allocation model not only discovers word correlations as topics (as LDA does), it also discovers topic correlations as super-topics. The Pachinko Allocation model generalises upwards to capture multiple levels of topic correlations. The hierarchical structure Pachinko Allocation is reminiscent of Pachinko machines[2] which gives rise to the name.

In [5], inference for Four-Level Pachinko Allocation is achieved via Gibbs sampling: a powerful, but a computationally intensive inference process. We have seen that Online Variational Bayes is a useful extension to LDA, yet we are not aware of an example of Online Variational Bayes for Pachinko Allocation although [18] alludes to a Batch Variational Bayes algorithm for Four-Level Pachinko Allocation.

2.5 Conclusions

We identify three key gaps in the literature in the context of mathematical document classification:

- By nature, mathematics consists of both text and mathematical notation. However, current approaches to mathematical document classification do not sufficiently explore the use of both textual and notational content.
- The current single vocabulary based approaches to mathematical document classification focus on traditional discriminative classifiers. Current approaches do not sufficiently explore the probabilistic structure of mathematical documents, nor do they account for the possibility of partially labelled data.

[2] a traditional Japanese game, in which balls are dropped and navigate through an arrangement of pins until landing in various bins at the bottom

- The LDA model is a very powerful tool for document classification, notably when equipped with Online Variational Bayes. Pachinko Allocation, however, is an interesting generalisation of LDA, but we have seen no examples Online Variational Bayes to provide fast inference.

To summarise, the current approaches to both general and mathematical document classification do not sufficiently address our problem. Our contribution is to investigate classifying mathematical documents by content, using both words and symbols, in a principled way, while presenting computationally efficient methods of doing so.

Chapter 3

Experimental Set-up

The primary claim to be tested in this thesis is that accurate mathematical document classification requires both textual and notational content. In this chapter, we introduce the experimental set-up we use to validate this claim.

We start with a corpus of partially labelled mathematical documents where the authors of the documents provide subject areas labels according to the Mathematics Subject Classification Scheme [19]. With this data, we encounter the *author-labelling problem*: authors are inconsistent with when it comes to the level of labelling they provide which presents a significant problem in multi-label classification. We describe this issue in more depth in Section 3.2, and also how some of the previously criticised preprocessing in [12] may be used to circumvent this concern in the context of performance evaluation.

To perform multi-labelled classification, we construct an ensemble of binary classifiers, one for each subject area. Similar to [13], we prevent over-fitting by partitioning the data into per-class partitions where we balance the amount of positive and negative training examples in each partition to equal numbers and the negative examples are drawn uniformly from all negative categories.

We then outline the general machine learning framework, in particular, classification via general latent topic models and discriminative classifiers, and finally, we describe the complete experimental set-up: the structure of each experiment and performance evaluation.

3.1 NTCIR Dataset

The data used for our experiments comes from the NTCIR dataset [20]. This dataset consists of a collection of approximately 104,000 scientific documents obtained from the arXiv converted to MathML, an XML-based document format which we describe in detail in Section 2.2. We choose to use this dataset for our research because the XML structure [21] of the document allows us to extract the mathematics from the relevant MathML tags easily.

The NTCIR dataset is also endowed with useful meta-data. In particular, approximately 35,000 documents are author labelled according to Mathematics Subject Classification scheme [19]. We describe the Mathematics Subject Classification labelling system in detail in Appendix A.2.

3.2 The Author Labelling Problem

In this section, we outline the difficulties of using author labelled documents in our machine learning framework. We begin with a comment made by an editor at Mathematical Reviews (MR) [22], an online database which contains reviews, bibliographic information, and mathematics subject classifications.

Each item in the MR database receives precisely one primary classification, which is the MSC code that describes its principal contribution. When submitting a document for review at MR, authors are asked for suggestions for the appropriate MSC codes which are also taken into consideration.

"An author may claim, or have pretensions, to be working in some areas whose relevance is only clear to them. On the other hand, at Mathematical Reviews there was a range of styles in assigning MSC codes; one editor tended to feel if there were a code in one of his areas that were just fine and quite enough; others heeded the call to express through codes all the subjects where there was significant work or relevance."

- Patrick Ion, Mathematical Reviews.[1]

[1]Personal communication

By nature, the author provided labels of the documents in the NTCIR dataset introduce both subjective labelling as well as inconsistent levels of labelling between documents. Such inconsistencies will have a significant impact on classification quality. Since imposing a non-subjective labelling system is infeasible, we seek alternative methods to avoid these problems. MR impose an editorial system to alleviate this, but inconsistencies are still evident.

To address the issues of the quality of the author labelling of the NTCIR dataset, we attempt to alleviate some of the inconsistencies between the authors by also performing our experiments on a uni-labelled version of the data. We arrange our data similar to [12]: we train our classifiers only on documents labelled with *exactly one* of the twenty most common top-level MSC codes. We impose one key difference: instead of discarding the remaining documents, we treat them as unlabelled and use these in the unsupervised layer of the machine learning framework which we describe later. This process leaves us with approximately 15,000 labelled documents.

Of course, the use of the uni-labelled dataset does not completely address the author labelling problem. Even though the level of labelling will be consistent throughout the dataset, some of the documents may be under-labelled, for example, an author may provide only one label when multiple labels are more appropriate. Furthermore, authors could simply mislabel some documents.

3.3 Document Level Preprocessing

In this section, we outline the document level preprocessing steps of the XML documents contained in the NTCIR dataset. In particular, we outline word and symbol feature extraction and filtering methods.

Feature Extraction

Firstly, we extract the words and symbols from the documents. We identify words as text within the document which does not belong to a `<math>` tag. We identify symbols as text which belongs to the either one of the MathML *mathematical operator* tags `<mo>`, or the MathML *mathematical identifier* tags `<mi>`.

Noise Filtering

We perform standard text pre-processing techniques to remove and group various words from the data [23]. To remove unnecessary words, we use the Natural Language Tool Kit package for Python [24] to remove a standard list of stopwords such as "and", "of", and "the". We then stem the words using the Snowball stemming algorithm [25] to collect words which share the same root, such as "solve", "solves", and "solving". Finally, we compile word and symbol vocabularies and filter out any extreme tokens: words and symbols which appear in more than 50% of the total number of documents or appear in less than five documents in the corpus

3.4 Data Partitioning

We introduce a data partitioning process which ensures that the classifiers are trained appropriately on the characteristics of both positive and negative instances of each class. This process prevents the classifiers being influenced by a biassed set of negative examples [26].

We outline a data partitioning process briefly alluded to in [13], which yields balanced training and testing partitions for each class, each with sufficient numbers of positive examples, balanced uniformly with negative examples of the other classes. Figure 3-1 shows the overall structure of a balanced partition and outline the full partitioning process below.

1. Separate the labelled documents in the corpus and set aside all unlabelled documents for unsupervised training.

2. For each class:

    - Separate the positive and negative examples for this class.
    - Randomly select 10% of the positive examples and tag as test documents for this class.
    - For each of these documents, tag this also as test documents for any negatively labelled classes. This step prevents too many positive examples being set aside for testing.

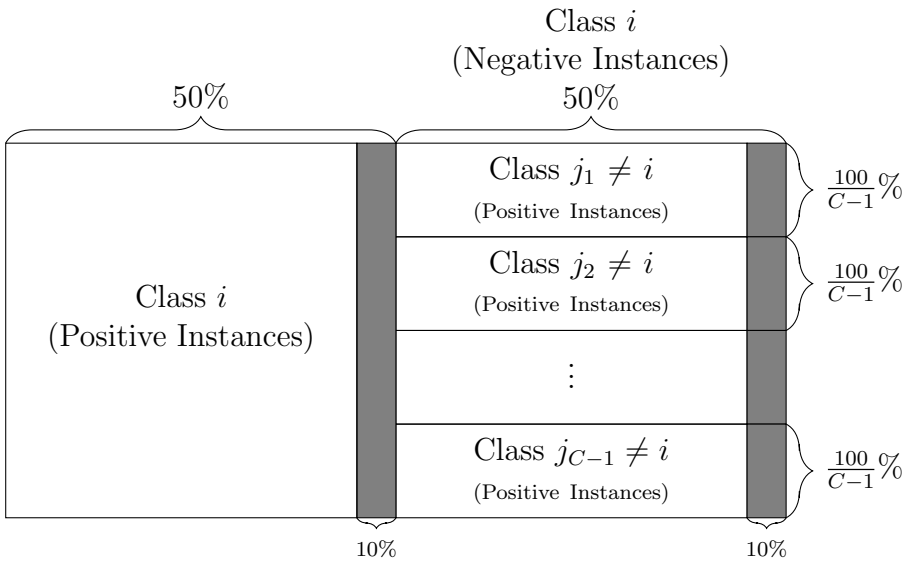

Figure 3-1: Structure of a balanced partition for the ith class out of C. The sub-partitions (shaded) are used for testing.

- Balance the number of positive and negative examples by adding or removing negative examples of this class maintaining uniformity across all other classes where possible.

3. Set aside all documents tagged for testing. We do not use these documents for training.

4. For each class:

   - Collect the remaining positive examples for this class and tag these as training documents for this class
   - Collect the same number of negative examples balanced uniformly across all other classes and also tag these as training documents for this class.

Note that these per-class training partitions are not necessarily disjoint; documents can appear as positive and negative examples in multiple partitions. As a result, the final partitions may not be perfectly balanced but with a sufficiently large dataset, this becomes insignificant.

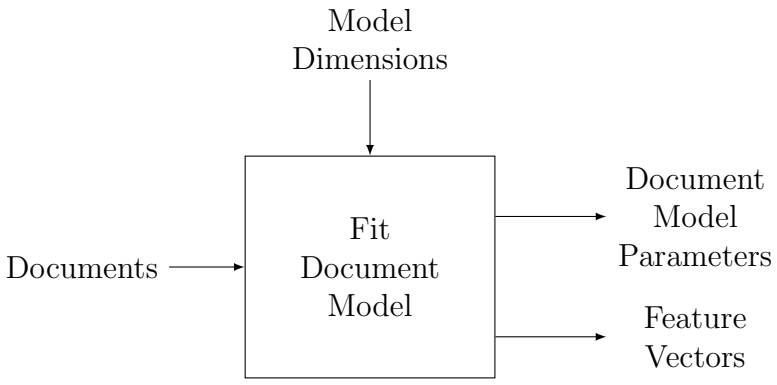

Figure 3-2: Structure of the Unsupervised Layer of the Machine Learning Framework

3.5 Machine Learning Framework

In this section, we describe the machine learning framework of our experimental set-up. The machine learning framework consists of two layers. Firstly, the *unsupervised layer* is the document modelling step which transforms a corpus of both labelled and unlabelled documents into a collection of feature vectors. Secondly, the *supervised layer* trains a supervised classifier using the labelled documents via these feature vectors.

Unsupervised Layer

The *unsupervised layer* of our classifier is responsible for feature extraction. In particular, given a corpus of documents, we wish to determine a small set of attributes that we can use as a basis for document representation. We require that this feature representation must preserve document similarity; similar documents regarding subject area should have similar feature vectors.

Formally, this step creates the *document model* of our corpus: given a corpus of documents and any dimensional information the model requires, yields a set of parameters which describe the model, and the corpus as feature vectors. This model should be unsupervised; this model should not require the document labels of a corpus. Figure 3-2 provides a diagrammatic illustration of the structure of the unsupervised layer.

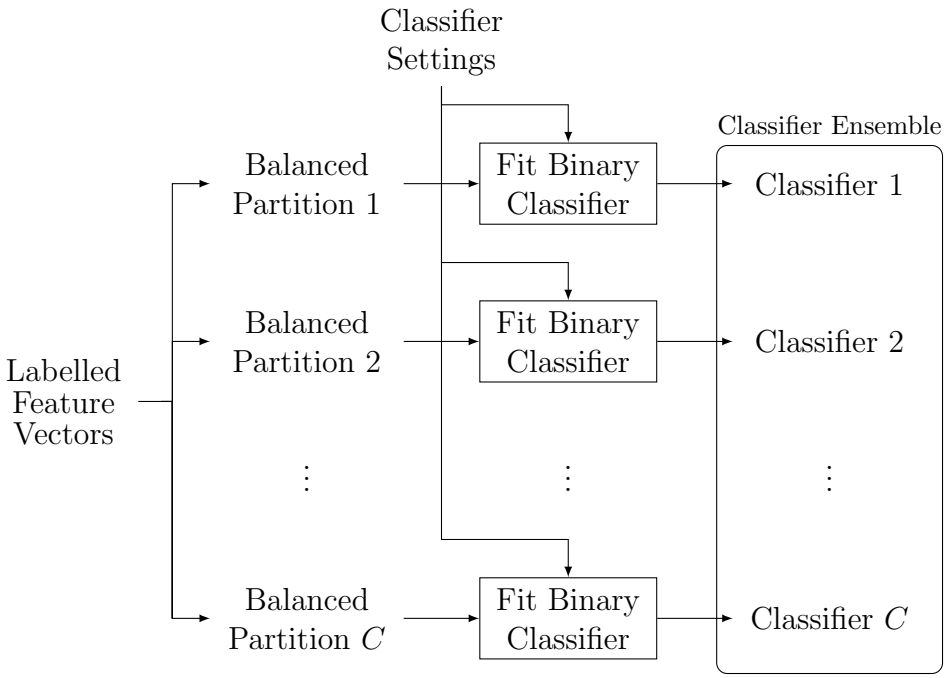

Figure 3-3: Structure of the Supervised Layer

Supervised Layer

Once we have a collection of labelled feature vectors, we can train an ensemble of binary classifiers to classify a document as positively or negatively belonging to each class. To do so, we first create balanced training partitions containing positive and negative examples of each document class. Secondly, given a balanced training partition, we create a binary classifier for the respective document class. Finally, the ensemble of each of these classifiers becomes the required multi-labelled classifier. Figure 3-3 provides a diagrammatic illustration of the supervised layer.

The previous sections outline the components of the machine learning framework used in this thesis. In particular, the design generalised to allow the utilisation of any document model in the unsupervised layer, and the use of any collection of binary classifiers in the supervised layer. By making adjustments to the partitioning process, we may replace the ensemble of classifiers with any multi-label classifier. Figure 3-4 shows the flow of documents in the machine learning framework process, in particular how the unsupervised layer and the

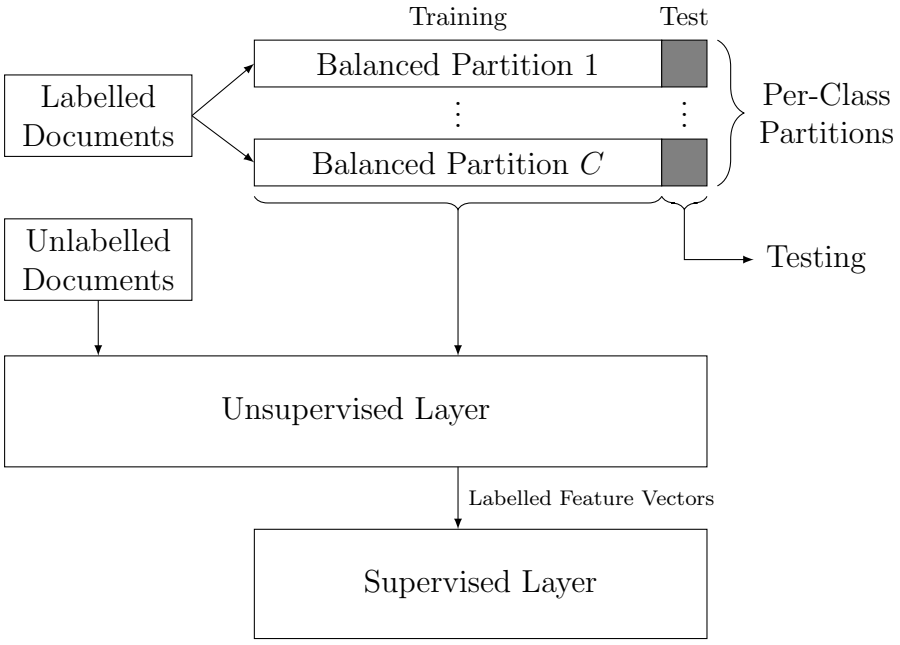

Figure 3-4: The flow of documents in the machine learning framework. The test documents are separated from the training partitions and not sent to the unsupervised layer. Only the labelled mixtures are passed to the supervised layer.

supervised layer fit together.

Document Classification

In this section, we outline the classification process of a document. By completing the training process, we obtain the following:

- The document model parameters which are needed to model an unseen document as a feature vector from the unsupervised layer.
- An ensemble of binary classifiers which predict the labels for a previously unseen document using the supervised layer.

The classification process is straightforward: to classify an unseen document, we represent it as a feature vector according to the document model obtained in the unsupervised layer and then run this feature vector through the classifier obtained from the supervised layer. Each binary classifier will return 1 if it predicts that

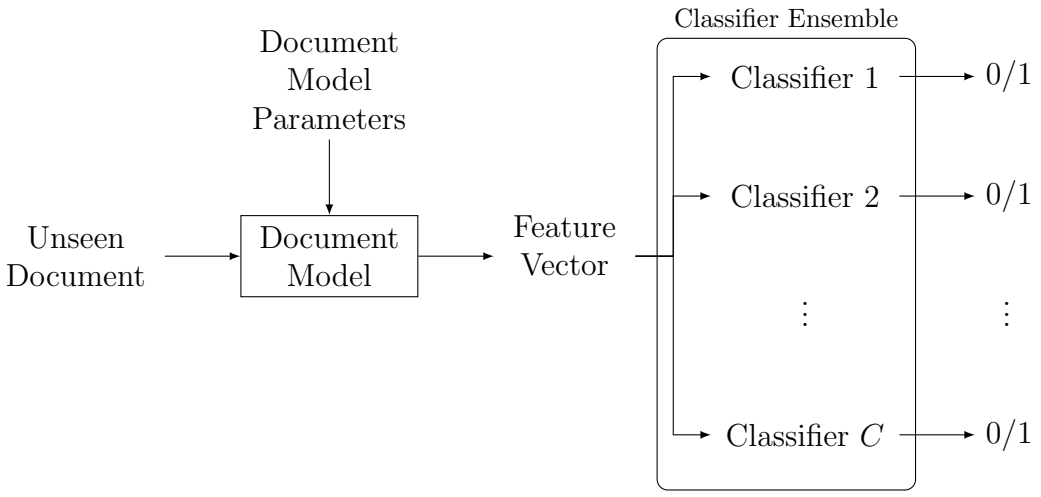

Figure 3-5: Structure of the document classification process. Each binary classifier returns 0 or 1 corresponding to the prediction of the document belonging to its respective class.

this feature vector (and, in turn, the document) belongs to that respective class, and 0 otherwise. Figure 3-5 shows a diagrammatic illustration of the document classification process.

3.6 Outline of Experiments

Chapters 4 to 7 describe four document models on which we perform our experiments. We have two document models over single vocabularies: the word-only models Latent Dirichlet Allocation in Chapter 4, and Pachinko Allocation in Chapter 6; and two document models over dual vocabularies: Dual Latent Dirichlet Allocation in Chapter 5, and Dual Pachinko Allocation in Chapter 7. We control each of these models with a selection of model dimensions of which we try various combinations for each experiment. Each experiment consists of a train/test phase:

1. Data partitioning: randomly partition the data as described in Section 3.5.

2. Unsupervised layer: fit the document model on the training partitions.

3. Supervised layer: train the per-class binary classifiers on the labelled feature vectors returned by the unsupervised layer.

4. Attempt to classify the documents in the test partition and measure classification performance using the performance measures which we describe in Appendix C.3.

We repeat each experiment sixteen times using different random partitions of training/test data and compare the distributions of the classification performance.

We perform the experiments in this thesis are on "Balena" [27], the High-Performance Computing Facility at the University of Bath on compute nodes with the specifications outlined in Table 3.1. Each node of Balena has 16 cores, so each experiment of 16 repetitions can be run in parallel on a single node in parallel.

System	Dell PowerEdge C8220
CPU	2X Intel E5-2650 v2 (20M Cache, 2.60 GHz)
Memory	64 GB DDR3-1866 MHz (8GB X 8)
OS	Scientific Linux release 6.5 (Carbon)
No. of Nodes	88

Table 3.1: Specifications of Balena

Chapter 4

Latent Dirichlet Allocation

The primary objective of this research is to create a dual vocabulary generative latent topic model for mathematical corpora. We begin by studying the *Latent Dirichlet Allocation* model (LDA) presented in [4], with the intention to adapt and refine this single vocabulary model to operate on a dual vocabulary. We emphasise that the *methods* outlined in this chapter are not novel; indeed, LDA is a popular tool for classification problems[1]. The novelty of this chapter is that we explore the applications of LDA in the context of mathematical document classification. In particular, we outline the methods in this chapter so as to understand the model in detail and study the approaches to inference and classification with the intention of using these as a basis of the original work which we present in following chapters.

Latent Dirichlet Allocation is a probabilistic generative model for collections of discrete data. In the context of document modelling, LDA assumes that each document can be represented by a random mixture of latent topics, where these topics are each characterised by a distribution of words. In particular, LDA attributes each word in a document to exactly one of these latent topics. [4] describes efficient inference techniques based on variational methods and an EM algorithm which approximates these latent topics from a corpus of documents. This inference process is further extended in [6] which outlines a significantly faster variant.

The topic mixture representation of a document according to LDA is a pow-

[1] At the time of print, Google Scholar (`https://scholar.google.co.uk`) reports over 15,000 citations for [4]

erful feature representation. Modelling documents as collections of word features requires classification over a very rich, but vast feature space [28]. In contrast, the topic mixture representation provides a significant dimension reduction to the feature space and still preserves document similarity; we may easily classify in this low-dimensional space instead.

To summarise, in this chapter, we first provide a detailed outline of the Latent Dirichlet Allocation model as described in [4], and furthermore, the Online Variational Bayes method presented in [6]. Secondly, we describe document classification via LDA, and finally, we perform various sets of experiments and discuss classification performance on our mathematical corpora.

4.1 Statistical Background

The LDA model is based heavily on the Categorical and Dirichlet probability distributions. In this section, we briefly outline these distributions and some of the key properties required for the mathematics in this and following chapters.

The Categorical Distribution

The Categorical distribution is the special case of the Multinomial distribution with only one trial [29]: given the number of categories $K \geq 2$, and a probability vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]$, the Categorical distribution models the set of indicator vectors $\mathbf{z}$ which index one of K possible outcomes, where the probability of each outcome is specified by the corresponding entry in $\boldsymbol{\theta}$. The probability density function of the Categorical distribution is given by

$$p(\mathbf{z}|\theta) = \prod_{i=1}^{K} \theta_i^{z_i} \tag{4.1}$$

and furthermore, the entries of the expected value of $\mathbf{z}$ is given by

$$\mathbb{E}[z_i|\boldsymbol{\theta}] = \theta_i \tag{4.2}$$

Notation

It is usually more convenient in calculations to express $\mathbf{z}$ as an indicator vector as above, however, in some calculations, it is more useful to have an integer representation corresponding to the non-zero entry (usually when we wish to use $\mathbf{z}$ to index a row in a matrix). In this case, we will use z (non-boldface) to denote the corresponding integer label. For example, if $\mathbf{z} = [0, 0, 1, 0]$ then we define $z = 3$.

The Dirichlet Distribution

The Dirichlet distribution is a continuous generalisation of the multinomial distribution. In particular, given the number of categories $K \geq 2$ and concentration parameter $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_K]$ of positive real numbers, the Dirichlet distribution models the space of probability vectors of length K. The probability density function of the Dirichlet distribution is given by

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^{K} \theta_j^{\alpha_j - 1} \tag{4.3}$$

where B denotes the multivariate Beta function given by

$$B(\boldsymbol{\alpha}) = \frac{\prod_{j=1}^{K} \Gamma(\alpha_j)}{\Gamma\left(\sum_{j=1}^{K} \alpha_j\right)}$$

[30].

We make a note of two useful properties of the Dirichlet distribution. In particular, the expected value and the expected log of the entries in $\boldsymbol{\theta}$ which are given by

$$\mathbb{E}[\theta_j|\boldsymbol{\alpha}] = \frac{\alpha_j}{\sum_{j'} \alpha_{j'}} \tag{4.4}$$

$$\mathbb{E}[\log \theta_j|\boldsymbol{\alpha}] = \Psi(\alpha_j) - \Psi\left(\sum_{j'} \alpha_{j'}\right) \tag{4.5}$$

where Ψ denotes the digamma function; the first derivative of the log Gamma function.

4.2 Latent Dirichlet Allocation

We now describe in detail the Latent Dirichlet Allocation model presented in [4]. LDA is a probabilistic generative model, where in the context of document modelling, LDA assumes that a document can be represented by a mixture of latent topics where these latent topics are each characterised by a distribution of words. In particular, each word is attributed to a latent topic via a latent topic index. Moreover, LDA assumes that the documents are generated by first sampling a latent topic mixture, and then Categorically sampling words from topics proportional to the entries in this topic mixture.

Formally, this model assumes that the number latent topics K, and the size of the vocabulary V are both known and fixed, the lengths of each document N_d are Poisson distributed with parameters ξ, and that the K topics are Dirichlet distributed with smoothing parameter $\boldsymbol{\eta}$. We note that the Poisson assumption is not critical to anything that follows in this Chapter.

Given the Dirichlet prior $\boldsymbol{\alpha}$ on the topic mixtures, and topics $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$, LDA assumes the generative process of a document $\mathbf{w}_d$ outlined in Algorithm 1.

Algorithm 1 Generative process of a document under LDA

Sample topic mixture $\boldsymbol{\theta}_d \sim \text{Dir}(\boldsymbol{\alpha})$.
Sample number of words $N_d \sim \text{Poisson}(\xi)$.
for each of the N_d words **do**
  Sample topic index $j = z_{dn} \sim \text{Cat}(\boldsymbol{\theta}_d)$.
  Sample word index $\mathbf{w}_{dn} \sim \text{Cat}(\boldsymbol{\beta}_j)$.
end for

Under this generative process, given the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$, the joint distribution of a document $\mathbf{w}_d$ with the latent topic mixture $\boldsymbol{\theta}_d$, topic indices $\mathbf{z}_d$, and topics $\boldsymbol{\beta}$ is given by the product

$$p(\mathcal{H}, \mathbf{w}_d|\boldsymbol{\Theta}) = p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \prod_{j=1}^{K} p(\boldsymbol{\beta}_j|\boldsymbol{\eta}) \prod_{n=1}^{N_d} p(\mathbf{z}_{dn}|\boldsymbol{\theta}_d)\, p(\mathbf{w}_{dn}|\mathbf{z}_{dn}, \boldsymbol{\beta}) \tag{4.6}$$

where $\mathcal{H}$ denotes the set of latent variables, and $\boldsymbol{\Theta}$ denotes the set of model parameters. The factors $p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})$, $p(\boldsymbol{\beta}_j|\boldsymbol{\eta})$ are given by the probability density function of the Dirichlet distribution, $p(\mathbf{z}_{dn}|\boldsymbol{\theta}_d)$ is given by the probability density

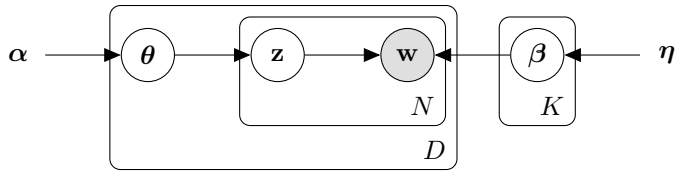

Figure 4-1: Graphical model representation of LDA

function of the Categorical distribution, and finally, the factors $p(\mathbf{w}_{dn}|\mathbf{z}_{dn}, \boldsymbol{\beta})$ are characterised by the probability density function of the Categorical distribution given by $p(\mathbf{w}_{dn}|\boldsymbol{\beta}_{z_{dn}})$.

Figure 4-1 illustrates the LDA model as a probabilistic graphical model [31]: a graph-like diagram where the directed edges denote dependencies between random variables and the plates denote the numbers of repeated nodes. Figure 4-1 makes clear the three levels of the model, in particular, we highlight the dependencies of:

1. The words on the word level topic indices.

2. The topic indices on the document level topic mixtures.

3. The topic mixtures on the corpus level Dirichlet prior.

Figure 4-2 shows an example of the topic mixtures for three documents and three topics under the LDA model.

4.2.1 Topic Mixture Representation

The LDA model allows us to characterise a document $\mathbf{w}_d$ as the proportions of the topic indices $\mathbf{z}_d$. In particular, this is the normalised frequency count of the topic indices given by

$$\bar{\mathbf{z}}_d = \frac{1}{N_d} \sum_{n=1}^{N_d} \mathbf{z}_{dn}$$

where the jth entry of $\bar{\mathbf{z}}_d$ represents the weight of topic j in the document.

The space of the possible topic mixtures $\bar{\mathbf{z}}_d$ is the $(K-1)$ dimensional probability simplex Δ^{K-1}: the space of non-negative $1 \times K$ vectors which sum to one. Figure 4-3 shows an example of the embedding of topic mixtures in the simplex Δ^2; the case when $K = 3$.

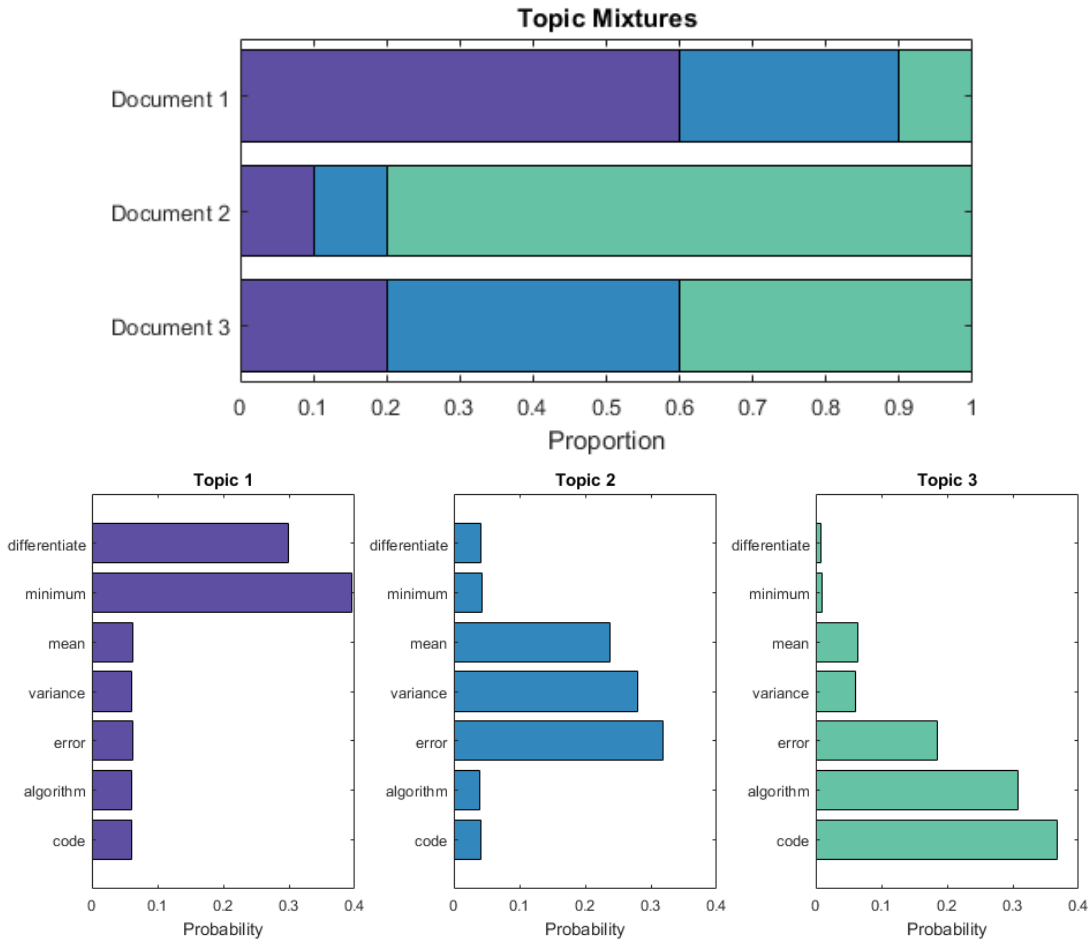

Figure 4-2: Example topic mixtures and corresponding topics under LDA.

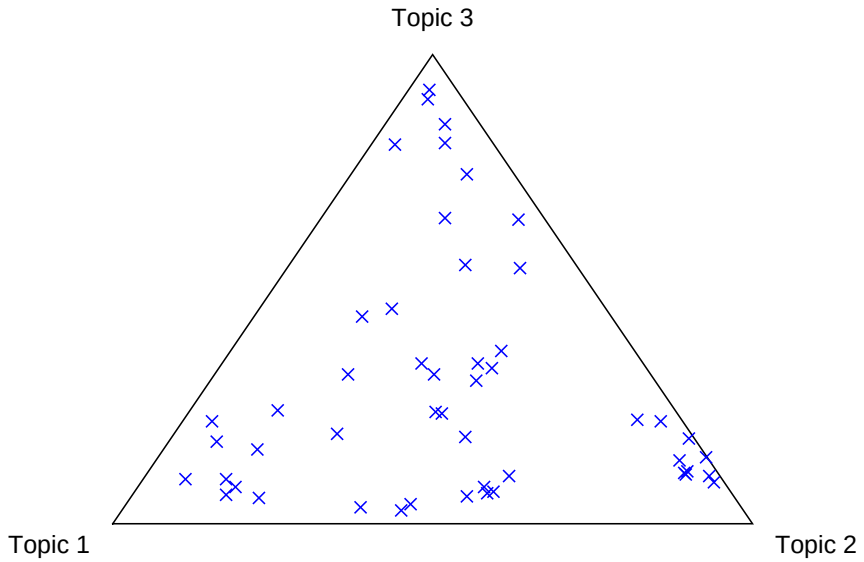

Figure 4-3: Example of documents embedded in the simplex Δ^2.

This geometric interpretation of the topic mixtures $\bar{\mathbf{z}}_d$ illustrates the concept of document similarity via topic mixtures: collections of similar documents (regarding subject area) form clusters in this space of topic mixtures. We use this notion as the basis for supervised document classification in which we describe in Section 4.4.

4.3 Inference

We now outline the process for solving for latent variables in the LDA model. Given the model parameters, we wish to determine the values of the latent variables which maximise the likelihood of a document $\mathbf{w}_d$ via the posterior distribution

$$p(\mathcal{H}|\mathbf{w}_d, \boldsymbol{\Theta}) = \frac{p(\mathcal{H}, \mathbf{w}_d|\boldsymbol{\Theta})}{p(\mathbf{w}_d|\boldsymbol{\Theta})}$$

This posterior distribution is intractable to compute in general since the marginal distribution $p(\mathbf{w}_d|\boldsymbol{\Theta})$ of a document expressed in terms of the latent variables is intractable to compute due to the interaction between $\boldsymbol{\theta}_d$ and $\boldsymbol{\beta}$ [32]. Instead, we employ a convexity-based variational algorithm to approximate this

posterior.

4.3.1 Variational Inference

We now outline the Batch Variational Bayes inference method as described in [4]. The general concept of convexity based variational inference is to use Jensen's inequality to obtain a lower bound of the log likelihood [33], which can then be adjusted to obtain the tightest lower bound. In particular, we consider a lower bound indexed by a set of free variational parameters, where these parameters can be optimised to find the tightest lower bound.

To achieve this, we consider a simpler version of the graphical model in Figure 4-1 and augment this model with a set of free variational parameters. In particular, we consider the graphical model outlined in Figure 4-4, where we retain only the nodes $\boldsymbol{\theta}$, $\mathbf{z}$, and $\boldsymbol{\beta}$ and introduce the free variational parameters $\boldsymbol{\gamma}$, $\boldsymbol{\phi}$ and $\boldsymbol{\lambda}$, with dependencies characterised by the variational distribution given by

$$q(\mathcal{H}|\mathcal{F}) = \prod_{j=1}^{K} q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j) \prod_{d=1}^{D} q_d(\mathcal{H}|\mathcal{F}) \tag{4.7}$$

where $\mathcal{F}$ denotes the set of free variational parameters, and q_d characterises the variational distribution of the dth document given by

$$q_d(\mathcal{H}|\mathcal{F}) = q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \prod_{n=1}^{N_d} q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn}) \tag{4.8}$$

In Equations (4.7) and (4.8), the factors $q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)$ and $q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j)$ are given by the probability density function of the Dirichlet distribution, and the factors $q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn})$ are given by the probability density function of the Categorical distribution.

Figure 4-4 makes clear the difference between the variational distribution and the full LDA model, and furthermore, highlights that a unique parameter governs each of the latent variables at the same level [34]. For example, the document level parameters $\boldsymbol{\theta}_d$ depend on document level variational parameters $\boldsymbol{\gamma}_d$ and not a corpus level parameter.

To obtain a lower bound on the log-likelihood of a corpus, we use the fact

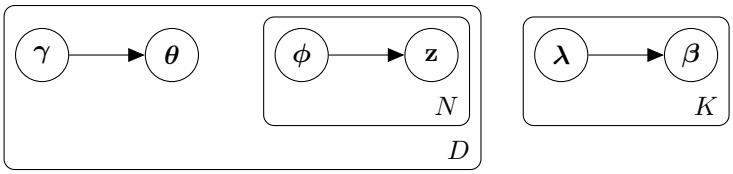

Figure 4-4: Graphical model representation of the variational distribution of LDA

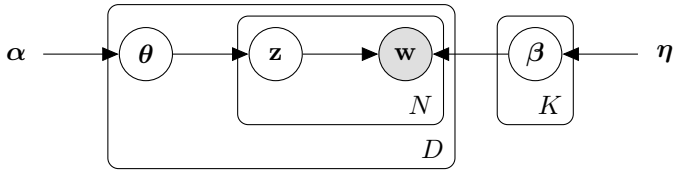

Figure 4-5: Graphical model representation of LDA

that $-\log$ is convex and employ Jensen's inequality[2]. We first marginalise over the latent variables and introduce the variational distribution q by expressing the log-likelihood as follows

$$\begin{aligned}\log p(\mathbf{w}|\boldsymbol{\Theta}) &= \log \iint \sum_{\mathbf{z}} p(\mathcal{H}, \mathbf{w}|\boldsymbol{\Theta}) d\boldsymbol{\theta}\, d\boldsymbol{\beta} \\ &= \log \iint \sum_{\mathbf{z}} \frac{p(\mathcal{H}, \mathbf{w}|\boldsymbol{\Theta}) q(\mathcal{H}|\mathcal{F})}{q(\mathcal{H}|\mathcal{F})} d\boldsymbol{\theta}\, d\boldsymbol{\beta}\end{aligned}$$

By applying Jensen's inequality to the right hand side of the above equation, we obtain a lower bound on the log-likelihood [33], and obtain the inequality

$$\begin{aligned}\log p(\mathbf{w}|\boldsymbol{\Theta}) \geq & \iint \sum_{\mathbf{z}} q(\mathcal{H}|\mathcal{F}) \log p(\mathcal{H}, \mathbf{w}|\boldsymbol{\Theta}) d\boldsymbol{\theta}\, d\boldsymbol{\beta} \\ & - \iint \sum_{\mathbf{z}} q(\mathcal{H}|\mathcal{F}) \log q(\mathcal{H}|\mathcal{F}) d\boldsymbol{\theta}\, d\boldsymbol{\beta}\end{aligned}$$

Finally, we note that this lower bound can be expressed in terms of expected values over the variational distribution q. In particular, we express the above inequality as

$$\log p(\mathbf{w}|\boldsymbol{\Theta}) \geq \mathbb{E}_q[\log p(\mathcal{H}, \mathbf{w}|\boldsymbol{\Theta})] - \mathbb{E}_q[\log q(\mathcal{H}|\mathcal{F})] \tag{4.9}$$

[2]For a random variable X and convex function f, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$.

We let $\mathcal{L}$ denote the right-hand side of Equation (4.9) as a function of $\mathcal{F}$ given the model parameters $\boldsymbol{\Theta}$ and call this the *Evidence Lower Bound*. We now rewrite $\mathcal{L}$ as a sum of expectations using the factorisations of p and q to obtain

$$\mathcal{L}(\mathcal{F}; \boldsymbol{\Theta}) = \sum_{d=1}^{D} L_d(\mathcal{F}; \boldsymbol{\Theta}) + \mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] - \mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] \tag{4.10}$$

where L_d, the contribution of the dth document to the Evidence Lower Bound, is given by

$$\begin{aligned} L_d(\mathcal{F}; \boldsymbol{\Theta}) = \mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] + \mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d)] + \mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})] \\ - \mathbb{E}_q[\log q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)] - \mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)] \end{aligned} \tag{4.11}$$

We now have a lower bound on the log-likelihood of the corpus given an arbitrary variational distribution q as required. We now wish to minimise the difference between the log-likelihood and this lower bound. It can be verified that this difference is the KL divergence[3] between the variational posterior probability $q(\mathcal{H}|\mathcal{F})$ and the true posterior probability $p(\mathcal{H}|\mathbf{w}, \boldsymbol{\Theta})$. Finally, by rewriting the log-likelihood in terms of the Evidence Lower Bound and this KL divergence, we obtain

$$\log p(\mathbf{w}|\boldsymbol{\Theta}) = \mathcal{L}(\mathcal{F}; \boldsymbol{\Theta}) + \mathrm{D}(q(\mathcal{H}|\mathcal{F}) \,\|\, p(\mathcal{H}|\mathbf{w}, \boldsymbol{\Theta}))$$

and we see that minimising the KL divergence between the variational posterior probability and the true posterior probability is equivalent to maximising $\mathcal{L}$ with respect to the free variational parameters. We now outline the steps for maximising the Evidence Lower Bound.

Expanding Expectations

To maximise the Evidence Lower Bound, we require the expectations in Equation (4.10) to be in terms of the variational parameters. In Appendix B, we outline the various forms that these expectations may take. Firstly, Appendices B.1 and B.2

[3]For continuous random variables P and Q, the Kullback-Leibler Divergence is given by $\mathrm{D}(P \,\|\, Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx$ where p and q denote the densities of P and Q respectively. [35]

show that the expectations over the topic mixtures $\boldsymbol{\theta}_d$ expand to the summations

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] = \log\Gamma\left(\sum_j \alpha_j\right) + \sum_{j=1}^{K}((\alpha_j - 1)\mathbb{E}_q[\log\theta_{dj}|\boldsymbol{\gamma}_d] - \log\Gamma(\alpha_j))$$
$$\mathbb{E}_q[\log q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)] = \log\Gamma\left(\sum_j \gamma_{dj}\right) + \sum_{j=1}^{K}((\gamma_{dj} - 1)\mathbb{E}_q[\log\theta_{dj}|\boldsymbol{\gamma}_d] - \log\Gamma(\gamma_{dj}))$$

where we emphasise the dependency on $\boldsymbol{\gamma}_d$ in the inner expectations. Appendices B.2 and B.3 show the expectations over the topics $\boldsymbol{\beta}$ expand to

$$\mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] = \sum_{j=1}^{K}\left(\log\Gamma\left(\sum_v \lambda_{jv}\right) + \sum_{v=1}^{V}((\lambda_{jv} - 1)\mathbb{E}_q[\log\beta_{jv}|\boldsymbol{\lambda}_j] - \log\Gamma(\lambda_{jv}))\right)$$
$$\mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] = K\left(\log\Gamma\left(\sum_v \eta_v\right) - \sum_{v=1}^{V}\log\Gamma(\eta_v)\right) + \sum_{v=1}^{V}(\eta_v - 1)\sum_{j=1}^{K}\mathbb{E}_q[\log\beta_{jv}|\boldsymbol{\lambda}_j]$$

where we emphasise the dependency on $\boldsymbol{\lambda}$ in the inner expectations. Appendices B.4 and B.5 show the expectations over the topic indices $\mathbf{z}_d$ expand and rearrange to give the summations

$$\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d)] = \sum_{n=1}^{N_d}\sum_{j=1}^{K}\phi_{dnj}\mathbb{E}_q[\log\theta_{dj}|\boldsymbol{\gamma}_d]$$
$$\mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)] = \sum_{n=1}^{N_d}\sum_{j=1}^{K}\phi_{dnj}\log\phi_{dnj}$$

where we emphasise the dependency on $\boldsymbol{\gamma}_d$ in the inner expectations. Finally, Appendix B.6 shows the expectations over the documents $\mathbf{w}_d$ expands and rearranges to give the summation

$$\mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})] = \sum_{n=1}^{N_d}\sum_{j=1}^{K}\phi_{dnj}\mathbb{E}_q[\log\beta_{jw_{dn}}|\boldsymbol{\lambda}_j]$$

where we emphasise the dependency on $\boldsymbol{\lambda}$ in the inner expectations.

By plugging the above expectations into Equation (4.10), we obtain the Evidence Lower Bound in terms of the model parameters and the free variational parameters. We now outline the process of maximising this lower bound via the

variational parameters $\mathcal{F}$.

4.3.1.1 Document Level Updates

In this section, we describe the methods of maximising $\mathcal{L}$ with respect to each of the document level variational parameters $\boldsymbol{\phi}_d$ and $\boldsymbol{\gamma}_d$.

Variational Categorical Parameters Firstly, we maximise $\mathcal{L}$ with respect to the Variational Categorical parameters $\boldsymbol{\phi}_{dn}$. We maximise each entry of $\boldsymbol{\phi}_{dn}$ individually and use Lagrange multipliers to enforce the constraint that the entries in $\boldsymbol{\phi}_{dn}$ must sum to one. Retaining only the terms of $\mathcal{L}$ containing ϕ_{dnj} and adding the appropriate Lagrange multiplier Λ_{dn} yields

$$\mathcal{L}_{[\phi_{dnj}]} = \phi_{dnj}(\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j] - \log \phi_{dnj}) + \Lambda_{dn}\left(\sum_{j'}\phi_{dnj'} - 1\right)$$

Taking partial derivatives of the above with respect to ϕ_{dnj} yields

$$\frac{\partial \mathcal{L}_{[\phi_{dnj}]}}{\partial \phi_{dnj}} = \mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j] - \log \phi_{dnj} - 1 + \Lambda_{dn}$$

Finally, using Equation (4.5) to expand the expected logs and setting this derivative to zero yields the maximising value of ϕ_{dnj} at

$$\phi_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\}$$

where we normalise $\boldsymbol{\phi}_{dn}$ so that the entries sum to one.

Variational Dirichlet We now maximise $\mathcal{L}$ with respect to Variational Dirichlet parameter $\boldsymbol{\gamma}_d$. We maximise each entry of $\boldsymbol{\gamma}_d$ individually with no constraints to enforce. Retaining only the terms of $\mathcal{L}$ containing γ_{dj}, we have

$$\mathcal{L}_{[\gamma_{dj}]} = \sum_{j'=1}^{K} \mathbb{E}_q[\log \theta_{dj'}|\boldsymbol{\gamma}_d]\left(\alpha_{j'} + \sum_{n=1}^{N_d}\phi_{dnj'} - \gamma_{dj}\right) - \log\Gamma\left(\sum_{j'}\gamma_{dj'}\right) + \log\Gamma(\gamma_{dj})$$

Taking partial derivatives of the above with respect to γ_{dj} yields

$$\frac{\partial \mathcal{L}_{[\gamma_{dj}]}}{\partial \gamma_{dj}} = \Psi'(\gamma_{dj})\left(\alpha_j + \sum_{n=1}^{N_d}\phi_{dnj} - \gamma_{dj}\right) - \Psi'\left(\sum_{j'}\gamma_{dj'}\right)\sum_{j'=1}^{K}\left(\alpha_{j'} + \sum_{n=1}^{N_d}\phi_{dnj'} - \gamma_{dj'}\right)$$

Finally, setting this derivative to zero yields the maximising value of γ_{dj} at

$$\gamma_{dj} = \alpha_j + \sum_{n=1}^{N_d}\phi_{dnj}$$

We now have the update rules for the variational parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$ which we require for the document level variational inference. Since these update rules for $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$ are dependent on one another, full variational inference requires alternating between these updates until convergence. We summarise the document level variational inference procedure in Algorithm 2.

Algorithm 2 Document level variational inference for Latent Dirichlet Allocation

Initialise $\boldsymbol{\gamma}_d$ randomly.
repeat
  Set $\phi_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\}$
  Set $\boldsymbol{\gamma}_{dj} = \alpha_j + \sum_n \phi_{dnj}$.
until Convergence of $\boldsymbol{\gamma}_d$.

Since the optimisation is conducted for a single fixed document $\mathbf{w}_d$, we may consider Algorithm 2 as a function of $\mathbf{w}_d$ that yields the optimised values for $\boldsymbol{\phi}_d$. In particular, we may approximate the topic mixture of a document as the expected value of the normalised frequency counts of the topic indices $\mathbf{z}_{dn}$ under the variational distribution q via the equation

$$\bar{\boldsymbol{\phi}}_d = \mathbb{E}_q[\bar{\mathbf{z}}_d] = \frac{1}{N_d}\sum_{n=1}^{N_d}\boldsymbol{\phi}_{dn}$$

where we use the fact that under the variational distribution q, the $\mathbf{z}_{dn}$'s are Categorically distributed with parameter $\boldsymbol{\phi}_{dn}$.

4.3.1.2 Corpus Level Updates

We now have a variational inference procedure for approximating the document level variational parameters $\boldsymbol{\gamma}_d$ and $\boldsymbol{\phi}_d$. To complete the full Variational Bayes process for LDA, we now derive the method for approximating the variational parameter $\boldsymbol{\lambda}$ and the Dirichlet prior α which maximise the log-likelihood of the data. We have already shown that there is a tractable lower bound on the log-likelihood, and we can further maximise this lower bound via the model parameter $\boldsymbol{\alpha}$ and the variational parameter $\boldsymbol{\lambda}$. Therefore, we can derive a full variational EM procedure that yields the optimised values for $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$.

Firstly, we optimise for $\boldsymbol{\lambda}$ using the same strategy as before; we maximise the lower bound $\mathcal{L}$ with respect to the individual entries λ_{jv}. Retaining only the terms of $\mathcal{L}$ containing λ_{jv} and simplifying, we have

$$\begin{aligned} L_{[\lambda_{jv}]} = \sum_{v'=1}^{V} \mathbb{E}_q[\log \beta_{jv'} | \boldsymbol{\lambda}_j] \left(\eta_{v'} + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{dnj} w_{dnv'} - \lambda_{jv'} \right) \\ - \log \Gamma \left(\sum_{v'} \lambda_{jv'} \right) + \log \Gamma(\lambda_{jv}) \end{aligned}$$

Using Equation (4.5) to express the inner expectations in terms of the variational parameters and taking partial derivatives with respect to λ_{jv} yields

$$\begin{aligned} \frac{\partial L_{[\lambda_{jv}]}}{\partial \lambda_{jv}} = \Psi'(\lambda_{jv}) \left(\eta_v + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{dnj} w_{dnv} - \lambda_{jv} \right) \\ - \Psi' \left(\sum_{v'} \lambda_{jv'} \right) \sum_{v'=1}^{V} \left(\eta_{v'} + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{dnj} w_{dnv'} - \lambda_{jv'} \right) \end{aligned}$$

Finally, setting this derivative to zero yields the maximising value of λ_{jv} at

$$\lambda_{jv} = \eta_v + \sum_{d=1}^{D} \sum_{n=1}^{N_d} w_{dnv} \phi_{dnj}$$

Note that this update for $\boldsymbol{\lambda}$ requires the full corpus of documents and the corresponding set of $\boldsymbol{\phi}$'s; this is the fundamental property of *batch* Variational Bayes.

Similar to estimating the expected value of $\boldsymbol{\theta}$'s from the $\boldsymbol{\gamma}$'s, we can find the expected value of $\boldsymbol{\beta}$ by taking the expectation value of $\boldsymbol{\beta}$ under the variational

distribution. That is, the jth topic can be approximated by

$$\hat{\boldsymbol{\beta}}_j = \mathbb{E}_q\big[\boldsymbol{\beta}_j\big] = \frac{\hat{\boldsymbol{\lambda}}_j}{\sum_v \lambda_{jv}}$$

We now maximise the lower bound $\mathcal{L}$ via the Dirichlet parameter $\boldsymbol{\alpha}$. Here we use the Newton-Raphson based method describe in [4] and [36] which requires a special structure of the Hessian of $\mathcal{L}$. Retaining only the terms of $\mathcal{L}$ containing $\boldsymbol{\alpha}$ we have

$$\mathcal{L}_{[\boldsymbol{\alpha}]} = \sum_{d=1}^{D}\left\{\log\Gamma\left(\sum_j \alpha_j\right) - \sum_{j=1}^{K}\log\Gamma(\alpha_j) + \sum_{j=1}^{K}(\alpha_j - 1)\mathbb{E}_q[\log\theta_{dj}|\boldsymbol{\gamma}_d]\right\}$$

Taking derivatives with respect to the Dirichlet component α_j gives the gradient

$$g_j = \frac{\partial\mathcal{L}}{\partial\alpha_j} = D\left(\Psi\left(\sum_{j'}\alpha_{j'}\right) - \Psi(\alpha_j)\right) + \sum_{d=1}^{D}\mathbb{E}_q[\log\theta_{dj}|\boldsymbol{\gamma}_d]$$

Finally, taking derivatives again with respect to Dirichlet component $\alpha_{j'}$ gives the Hessian

$$H_{jj'} = \frac{\partial^2\mathcal{L}}{\partial\alpha_j\partial\alpha_{j'}} = -\delta(j,j')D\Psi'(\alpha_j) + D\Psi'\left(\sum_{j''}\alpha_{j''}\right)$$

which is of the required form to apply the Newton-Raphson method described in [4] and [36]. In particular, [36] provides an algorithm for estimating $\boldsymbol{\alpha}$ via the update rule

$$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}(\boldsymbol{\gamma})$$

where $\tilde{\boldsymbol{\alpha}}$ is the inverse of the Hessian H multiplied by the gradient $\mathbf{g}$ as a function of the $\boldsymbol{\gamma}_d$'s.

We now have the required document level and corpus level updates needed for Batch Variational Bayes for Latent Dirichlet Allocation. These updates are guaranteed to converge to a stationary point of the Evidence Lower Bound [6], and we may partition the updates to give the Expectation-Maximisation (EM) algorithm [37]:

- E-step: For each document, find the optimised values of the variational

parameters $\boldsymbol{\gamma}_d$ and $\boldsymbol{\phi}_d$ using the document level variational updates.

- M-step: Maximise the resulting lower bound on the log-likelihood via the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ using the corpus level updates.

We summarise the full variational inference procedure on a corpus of documents in Algorithm 3. We apply the document level variational updates for each document in the corpus, and update the corpus level parameters after each pass of the data.

Algorithm 3 Batch Variational Bayes for Latent Dirichlet Allocation

Initialise $\boldsymbol{\lambda}$ randomly.
repeat
  for $d = 1, \ldots, D$ **do**
    Initialise $\boldsymbol{\gamma}_d$ randomly.
    repeat
      Set $\phi_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj} | \boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}} | \boldsymbol{\lambda}_j]\}$
      Set $\boldsymbol{\gamma}_{dj} = \alpha_j + \sum_n \phi_{dnj}$.
    until Convergence of $\boldsymbol{\gamma}_d$.
  end for
  Set $\lambda_{jv} = \eta_v + \sum_d \sum_n w_{dnv} \phi_{dnj}$.
  Update $\boldsymbol{\alpha}$ according to [36].
until Convergence of $\mathcal{L}$.

4.3.2 Online Variational Inference

Batch Variational Bayes (Algorithm 3) requires a full pass of the corpus at each iteration and therefore can be slow to apply to large corpora or situations where documents arrive in a stream. In this section, we outline Online Variational Bayes for Latent Dirichlet Allocation as described in [6], which requires only a single pass of the corpus and updates the Variational Dirichlet parameter $\boldsymbol{\lambda}$ after the observation of each document.

Online Variational Bayes is very similar to Batch Variational Bayes with only a few extra/modified steps. The key difference is that at each iteration, $\boldsymbol{\lambda}$ is updated using a weighted average of its previous value and $\tilde{\boldsymbol{\lambda}}$, the optimal value of $\boldsymbol{\lambda}$ give the current $\boldsymbol{\phi}_d$. The weight of $\tilde{\boldsymbol{\lambda}}$ at iteration d is given by $\rho_d := (\tau_0 + d)^{-\kappa}$.

The parameter ρ_d depends on two corpus level constants: κ (the decay) which controls the rate of with old values of $\tilde{\boldsymbol{\lambda}}$ are forgotten, and τ_0 (the offset) which slow down the initial iterations of the algorithm. Convergence is guaranteed when the condition $\kappa \in (0.5, 1]$ is satisfied.

We outline a new method as described in [6] for estimating the parameter $\boldsymbol{\alpha}$ which is very similar to the Newton-Raphson method in the batch scenario. We now update $\boldsymbol{\alpha}$ after each iteration using a modified version of the Newton-Raphson algorithm employed in the batch scenario where the update step for $\boldsymbol{\alpha}$ now incorporates the offset and delay parameters by replacing the following update rule which depends on the weighting parameter ρ_d

$$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \rho_d \tilde{\boldsymbol{\alpha}}(\boldsymbol{\gamma}_d)$$

where $\tilde{\boldsymbol{\alpha}}$ is a function of $\boldsymbol{\gamma}_d$ that yields the inverse of the Hessian H times the gradient g where

$$g_j = \Psi\left(\sum_{j'} \alpha_{j'}\right) - \Psi(\alpha_j) + \mathbb{E}_q[\log \theta_{dj} | \boldsymbol{\gamma}_d]$$
$$H_{jj'} = \Psi'\left(\sum_{j''} \alpha_{j''}\right) - \delta(j, j'')\Psi'(\alpha_j)$$

are functions of $\boldsymbol{\gamma}_d$. We describe full Online Variational Bayes for Latent Dirichlet Allocation in Algorithm 4.

Algorithm 4 Online Variational Bayes for Latent Dirichlet Allocation

Define $\rho_d := (\tau_0 + d)^{-\kappa}$

Initialise $\boldsymbol{\lambda}$ randomly

for $d = 1, 2, \ldots$ **do**

  Initialise γ_d randomly

  repeat

    Set $\phi_{dnj} \propto \exp\left\{\Psi(\gamma_{dj}) + \Psi(\lambda_{jw_{dn}}) - \Psi\left(\sum_v \lambda_{jv}\right)\right\}$

    Set $\boldsymbol{\gamma}_{dj} = \alpha_j + \sum_n \phi_{dnj}$

  until Convergence of $\boldsymbol{\gamma}_d$

  Set $\tilde{\lambda}_{jv} = \eta_v + D \sum_n w_{dnv} \phi_{dnj}$

  Set $\boldsymbol{\lambda} = (1 - \rho_d)\boldsymbol{\lambda} + \rho_d \tilde{\boldsymbol{\lambda}}$

  Update $\boldsymbol{\alpha}$ according to [6].

end for

4.4 Document Classification

In this section, we describe document classification via LDA. In particular, we describe a *semi-supervised* classifier; a partially labelled corpus is used to approximate the LDA topics and the labelled documents are used to train a supervised multi-label classifier on their corresponding topic mixture representations. Recall that given a document $\mathbf{w}_d$, the document level variational procedure outlined in Algorithm 2 yields an approximation of its associated topic mixture representation $\bar{\boldsymbol{\phi}}_d$.

We now outline the complete framework for building a document classifier based on using the topic mixture representations under the LDA model as feature vectors.

4.4.1 Framework

We use the machine learning framework described in Chapter 3. In particular, we break down the machine learning process into two layers. Firstly, the unsupervised layer which uses LDA document model to approximate the latent topics and determine the topic mixture representations of the documents. Secondly, the supervised layer which trains a supervised multi-label classifier on a collection of

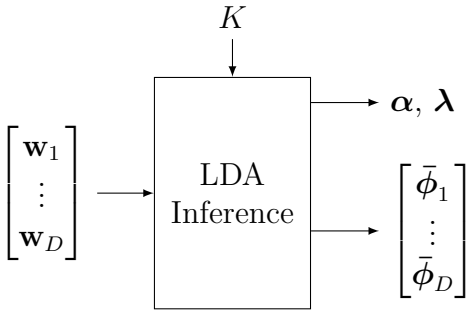

Figure 4-6: The Unsupervised Layer using LDA

labelled topic mixtures.

Document Modelling

Given a partially labelled corpus of documents and a choice of K, we use the LDA parameter estimation methods to approximate the Dirichlet prior $\boldsymbol{\alpha}$ on the topic mixtures, the variational parameter $\boldsymbol{\lambda}$ on the topics, and the topic mixture representations $\bar{\boldsymbol{\phi}}_d$ of the documents. Figure 4-6 shows a diagrammatic illustration of the document modelling step (the unsupervised layer) of the machine learning framework using LDA.

Supervised Training

By using the labelled topic mixture representations obtained from the unsupervised layer as feature vectors, we train a supervised multi-label classifier. We may use any supervised classification methods for this step; we simply require a function f such that $f(\bar{\boldsymbol{\phi}})$ yields the predicted classes of a topic mixture $\bar{\boldsymbol{\phi}}$. Figure 4-7 shows a diagrammatic illustration of the supervised layer using a collection of labelled topic mixtures from LDA.

In this thesis, we focus on nearest neighbour methods which we describe in detail in Appendix C.2. Recall, that the space of possible topic mixtures over K topics is the probability simplex Δ^{K-1}. We remind ourselves that the probability simplex is non-Euclidean thus traditional Euclidean distance metrics are not appropriate for evaluating the similarity between observations [38]. Instead, we must use a metric more suited for comparing histograms such as the χ^2 distance

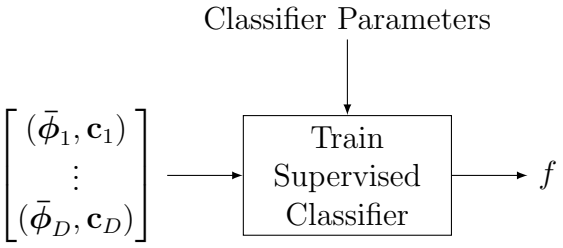

Figure 4-7: The Supervised Layer of the LDA classifier

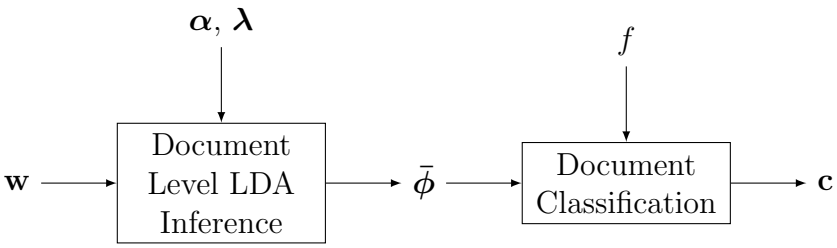

Figure 4-8: Classification process via LDA

metric [39]. The χ^2 distance between two points $\mathbf{x}$ and $\mathbf{y}$ of dimension K is given by

$$d(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^{K} \frac{(x_i - y_i)^2}{x_i + y_i}$$

where x_i and y_i denotes the ith entries of $\mathbf{x}$ and $\mathbf{y}$ respectively.

Classification

We now have all the components necessary for classification. The unsupervised layer provides the required parameters to obtain the topic mixture representation of an unseen document, and the supervised layer provides the document classifier f which will output the predicted set of labels for this document. Figure 4-8 outlines the classification process of a previously unseen document.

To summarise, LDA provides fast filtering algorithm for feature selection for document classification and furthermore, the topic mixture representation yields a significant dimension reduction compared to using possibly thousands of word features. Finally, training a supervised multi-label classifier on the labelled topic mixtures provides us with our desired document classifier.

4.5 Experimental Results

In this section, we evaluate the performance of mathematical document classification via LDA using the experimental set-up as described in Chapter 3. We first perform a set of preliminary experiments to get a feel of how classification via Latent Dirichlet Allocation behaves. In particular, we observe what effect the choice of the number of latent topics K has on classification performance. After identifying the optimal choice of K, we investigate the best performing classifiers in detail and study the confusion between subject areas.

4.5.1 Preliminary Experiments

For our first set of experiments, we fit the Latent Dirichlet Allocation model to our training data using the values of $K \in \{10, 25, 50, 100, 200, 250\}$. For the supervised layer, we train an ensemble of Nearest Neighbour Classifiers which we describe in Appendix C.2, using five nearest neighbours, inverse distance weighting and the χ^2 distance metric.

Effect of K

Figure 4-9 shows the results of our first set of experiments and highlights the effect that the choice of K has on classification accuracy via the Labelling F-Score performance measure [40]. We repeat each experiment sixteen times using different random test and train partitions of the data and plot the median Labelling F-Score of each collection of tests equipped with error bars highlighting the interquartile range.

The results of our preliminary experiments show that classification performance increases as K increases, with the best performance at $K = 200$ before we observe no increase in performance.

Precision/Recall Trade-off

We now check for an imbalance of false positive and false negative classification rates by investigating the micro-averaged precision and recall of this classifier. Across all sixteen experiments of the $K = 200$ case, we observe a median micro-averaged precision and recall of 89.00% and 82.52% respectively on the

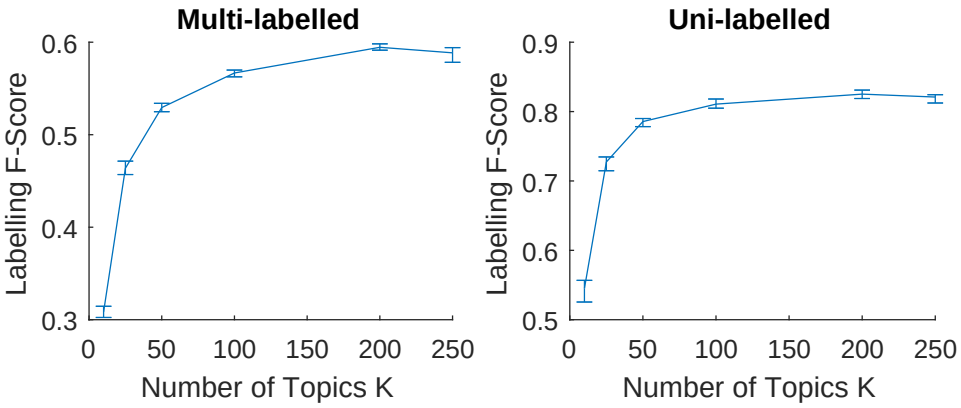

Figure 4-9: Classification performance via LDA

uni-labelled dataset, and 70.39% and 46.88% respectively on the multi-labelled dataset.

Since the classifiers on both datasets have high precision and relatively low recall, we conclude that the individual binary classifiers have some leeway when it comes to the confidence of a positive classification. In particular, allowing the classifiers to yield a positive result with slightly more uncertainty should, in theory, increase the overall classification performance.

The supervised layer of this model is an ensemble of nearest neighbour classifiers equipped with a centralised threshold. That is, the classifiers only yield positive classifications if the other positive examples are the *closest* to the new observation. In our application, it may be appropriate to yield positive classifications if there are a sufficient number of other positive examples which are indeed *close*, but perhaps not necessarily the closest.

We force a different balance of precision and recall by adjusting the decision threshold value of the binary classifiers. For example, if we set the threshold value to 100% (i.e. reject all) will yield high precision (no false positives), and a threshold value of 0% (i.e. accept all) will yield high recall (no false negatives). We discuss adapted decision thresholds in the K nearest neighbour setting in detail in Appendix C.2.

We run a new set of experiments for the $K = 200$ case on both datasets using different threshold values for the supervised classifiers. We discover that on both datasets, using a decision threshold of 0.3 yields optimal performance with

a Labelling F-Score of 65.14% and 87.35% on the multi-labelled and uni-labelled scenarios respectively. We outline the details of the optimal classifiers in Table 4.1 where we also report the median and maximum micro-averaged F-Score for the classifiers to directly compare to the work in [12] and [13] which we discuss further in Section 4.5.3.

	Multi-labelled	Uni-labelled
K	200	200
Threshold	0.3	0.3
Labelling F-Score		
(median)	64.5%	85.96%
(maximum)	65.14%	87.35%
Micro-averaged F-Score		
(median)	63.01%	86.08%
(maximum)	63.73%	87.75%

Table 4.1: Optimised Classification Performance via Online Latent Dirichlet Allocation

Remark

We note that here that we may have introduced an element of bias to the results. In particular, our selection of the best possible choice of K is dependent on the test data and not the training data. We do not believe this to be a major cause for concern; [4] reports strong classification using small partitions of the data for training. That is to say, by also setting aside a validation set to select the optimal value of K, which in turn reduces the size of the training set, we would not expect a decline in performance due to the smaller training set, and we would expect the classification results to remain similar on the validation set.

4.5.2 Confusion

In this section, we investigate the confusion between subject areas that our classifier exhibits. We select one of the sixteen experiments at random and look closer at the outputs of the classifiers directly and evaluate the confusion between classes by calculating the following three classification/misclassification rates:

1. The true positive rates of each class i: the percentages of condition positives of class i that are true positives.

2. The false negative rates of each class i distributed over each other class j: the percentages condition positives of class i which are both false negatives on class i and false positives on class j.

3. The null classification rates of each class i: the percentage of contrition positives of class i that have been predicted to belong to exactly zero classes.

We present this information in a confusion matrix where the entries on the diagonal correspond to the per-class true positive rates, and the ijth entry denotes the false negative rates described by item 2 above, and the final column corresponds to the rates of null predictions of each class. In the confusion matrices we present, we also impose a heat-map where the intensities of green and red correspond to the strengths of the rates of classification and misclassification respectively, and we sort the rows and columns via the true positive rates on the diagonal.

Firstly, Figure 4-10 shows the confusion matrix for the uni-labelled classification scenario. We observe good per-class true positive rates ranging from 100% to 60%. We see some interesting areas of confusion: we notice strong (and perhaps forgivable) confusion between pairs of strongly related subject areas, with the strongest confusion between the subject areas of "Statistics" and "Probability theory and stochastic processes".

We now construct the equivalent confusion matrix for the multi-labelled scenario. Due to the number of classes and the size of the confusion matrix, we present three smaller sections of the confusion matrix for readability and present the full heat-map in Appendix D. In particular, we show sections highlighting the strongest per-class true positive rates, the highest rates of confusion, and the weakest per-class true positive rates.

Since we are now dealing with multi-labelled data, we inherently introduce some noise to the confusion matrix. In particular, for each classification instance, we account each false positives against each false negatives in each classification. Here we make the assumption that the confusion distributed equally amongst classes, where in practice we would expect the "real" confusion to occur between smaller subsets of classes appearing in documents.

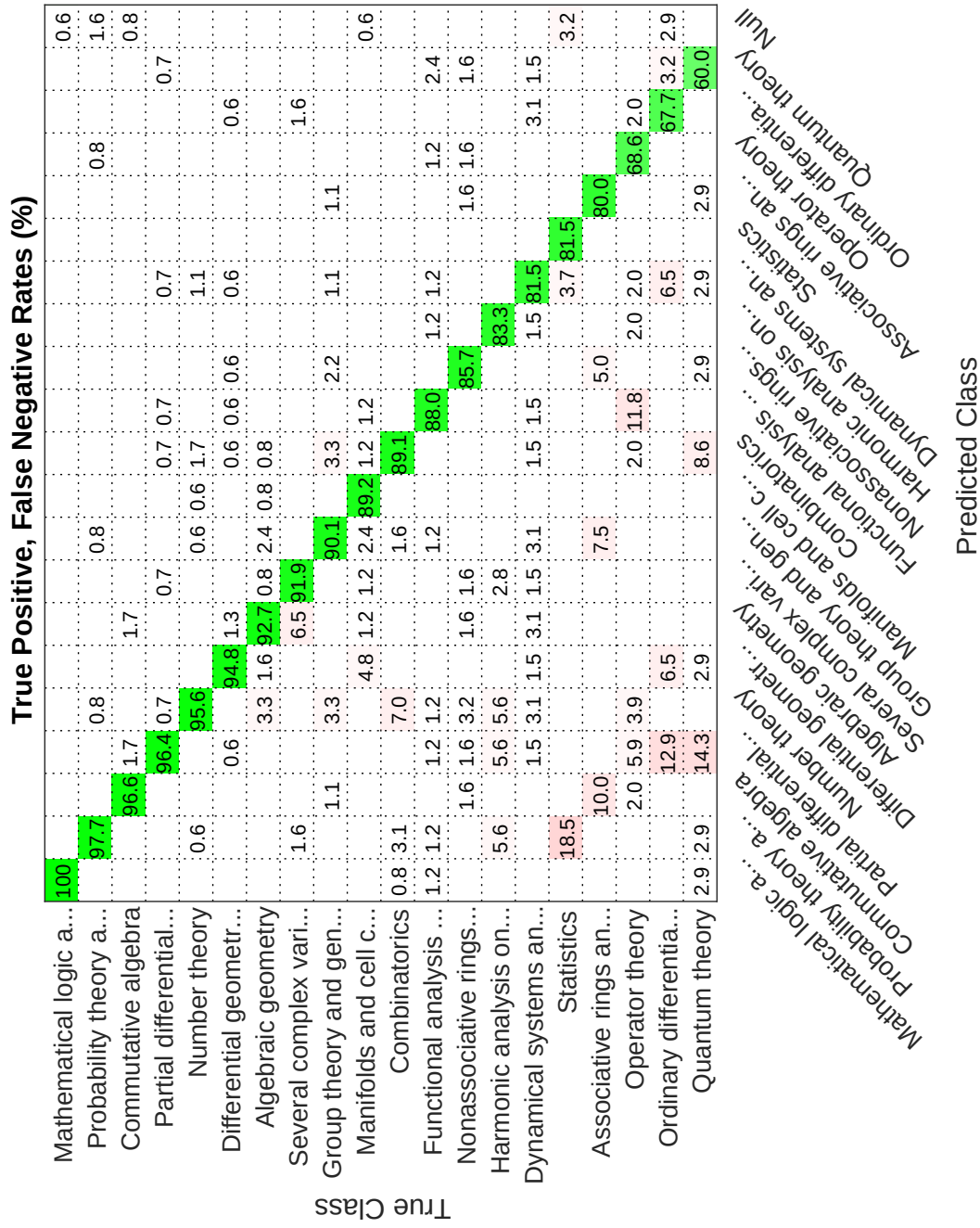

Figure 4-10: Per-class confusion of classification via LDA on the uni-labelled dataset

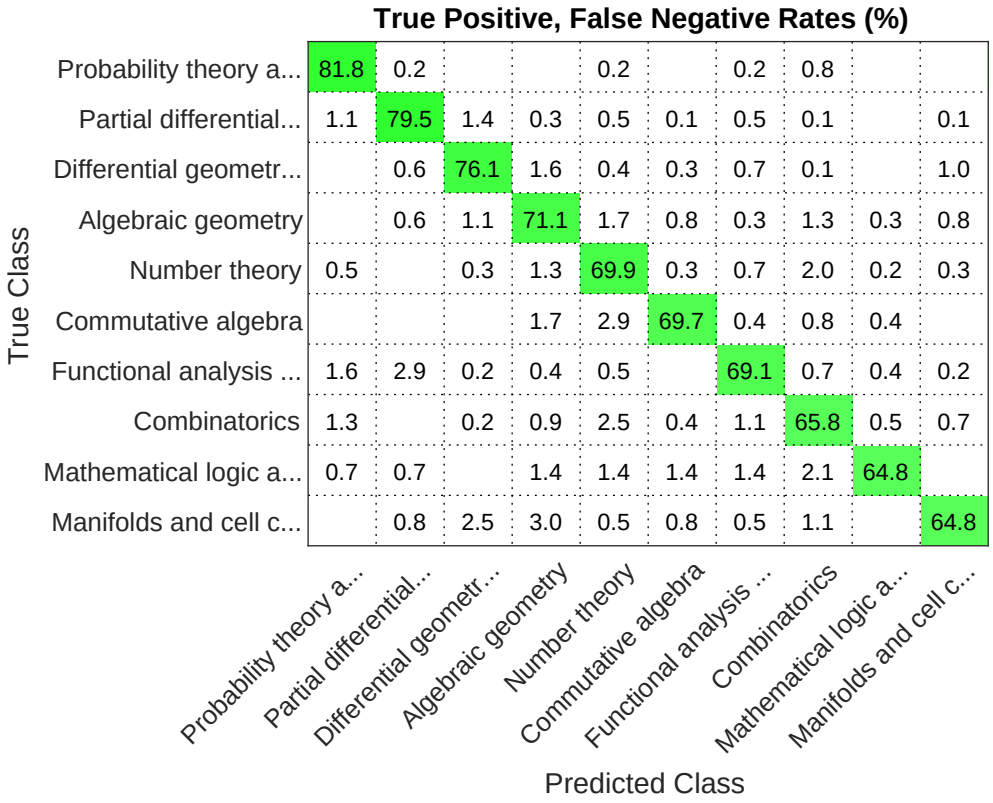

Figure 4-11: Per-class confusion of classification via LDA on the multi-labelled dataset - Upper-left section.

Firstly, Figure 4-11 shows the section of the confusion matrix highlighting the top ten strongest per-class true positive rates; the upper-left section of the confusion matrix. Here we see reasonably strong true positive rates with very slight confusion between almost all pairs of classes.

We now look at some the sections highlighting the areas of weak classification. Figure 4-12 shows the area of the confusion matrix which highlights the strongest levels of confusion between subject areas; the lower-left section of the confusion matrix. Similar to the uni-labelled scenario, we notice the strongest confusion between the strongly related subject areas. This time, we notice the strongest confusion between "Integral equations" and "Optics, electromagnetic theory" with "Partial differential equations".

Finally, Figure 4-13 shows the section of the confusion matrix highlighting the weakest per-class true positive rates; the lower-right section of the confusion

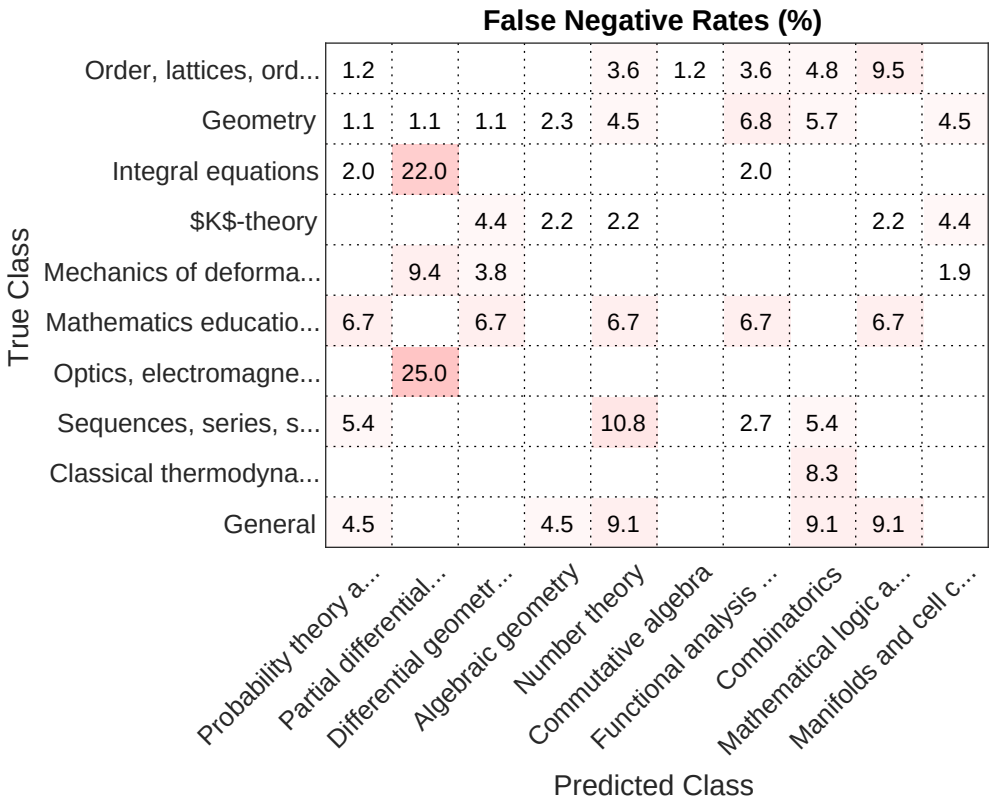

Figure 4-12: Per-class confusion of classification via LDA on the multi-labelled dataset - Lower-left section.

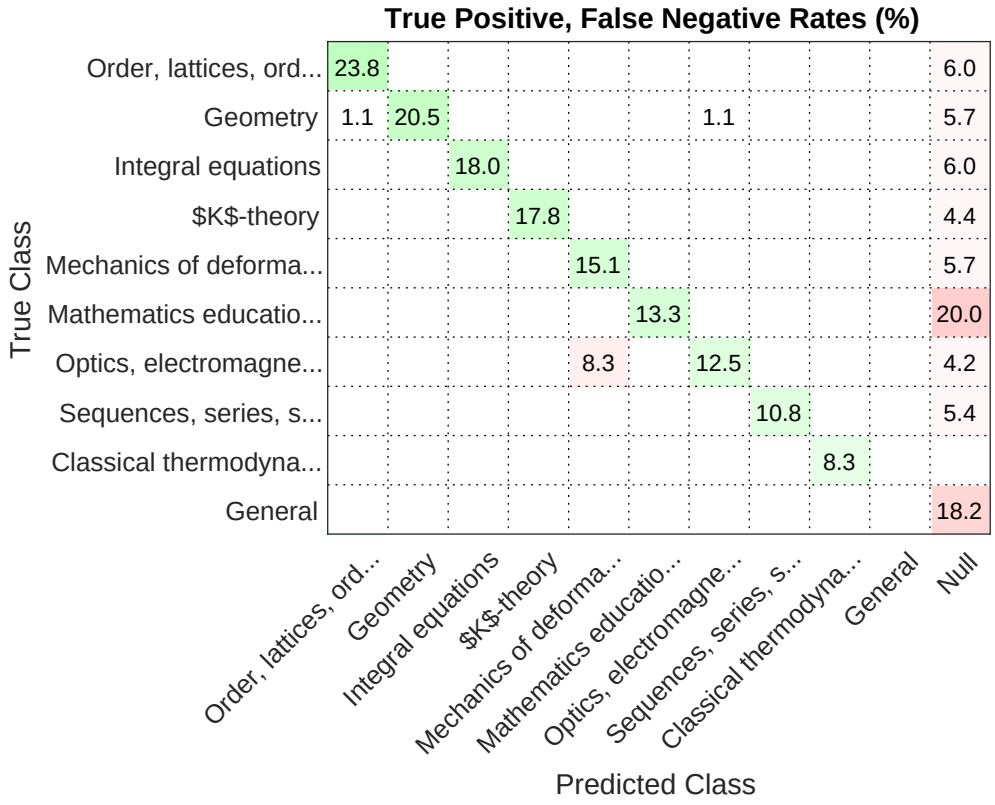

Figure 4-13: Per-class confusion of classification via LDA on the multi-labelled dataset - Lower-right section.

matrix. Here we see that the poorest per-class classification performance occurs within the more generic subject areas such as "General" and "Mathematics education" which also exhibit the highest rates of null classifications. Furthermore, we see poor performance on subject areas which are closer to applied sciences such as "Classical thermodynamics, heat transfer", "Optics, electromagnetic theory" and "Mechanics of deformable solids".

4.5.3 Discussion

The results here look promising. The classifiers are performing well on both datasets and are comparable in performance with [12] and [13]. We now summarise our findings.

On both datasets, we observe that as the choice of K (the number of latent

topics to be discovered) increase, the classification performance also increases until approaching the $K = 200$ mark. On further investigation, we see that using a decision threshold of 0.3 on the Nearest Neighbour binary classifiers yields overall better classification performance over a centralised decision threshold. Finally, using the optimal values of K and the decision threshold, we observe a relatively high median Labelling F-Score of 64.50% when classifying the multi-labelled dataset which suggests that classification via LDA is a strong contender for the real world setting of mathematical document classification. The highest micro-averaged F-Score we observe is 63.73% which is comparable to the results in [13] which reports a maximum micro-averaged F-Score of 67.3%.

Looking at the uni-labelled classification results, we observe a high median Labelling F-Score of 85.96% which suggests that the per-document classifications are consistently accurate across the top twenty classes. The highest micro-averaged F-Score we observe is 87.75% which is again comparable in performance to the results in [12] which reports a maximum micro-averaged F-Score of 89.03%.

To conclude, Latent Dirichlet Allocation proves itself to be a strong tool for mathematical document classification yielding comparable performance to the current state of the art. We now aim to improve upon these results by using LDA as a base model and introducing mathematical symbol data in the classification process. In the next chapter, we introduce Dual Latent Dirichlet Allocation: our refinement of this model which accounts for observations spanning two separate vocabularies and allows us to model mathematical documents as collections of both words and symbols.

Chapter 5

Dual Latent Dirichlet Allocation

In the previous chapter, we introduce Latent Dirichlet Allocation, a powerful latent topic model which, in the context of document modelling, characterises the correlations between words as latent topics. LDA, however, does not directly address our problem of mathematical document classification where we assume observations over a dual vocabulary. In this chapter, we present Dual Latent Dirichlet Allocation (DLDA): a novel generalisation of the LDA model which observes mathematical documents over a dual vocabulary.

The novelty here is that DLDA makes the same assumptions as the LDA model, and further assumes that each symbol is attributed to a latent symbol topic. Under DLDA, these topics are each characterised by both a distribution of words and a distribution of symbols. Similar to LDA, we solve for latent variables in the DLDA model using Online Variational Bayes.

To create the Dual Latent Dirichlet Allocation model, we take inspiration from the Gaussian-Multinomial Latent Dirichlet Allocation model (GM-LDA) described in [41], which is used to model annotated images. In particular, GM-LDA is used to model collections of "caption" words (the LDA part) and corresponding descriptors of image regions (the Gaussian-Multinomial part). Under GM-LDA, each word and image feature is attributed to a latent word and image topic respectively. The word and image topics here are characterised by Categorical distributions and *Gaussian* distributions respectively. The DLDA model presented in this chapter is of similar structure to GM-LDA, but where the two sets of topics are both characterised by Categorical distributions.

To summarise, we begin by providing a detailed outline of the Dual Latent

Dirichlet Allocation model: we outline the generative process, the Online Variational Bayes procedure, classification via DLDA. Finally, we perform a set of experimental tests and compare to our results to classification via LDA and the work in [12] and [13].

Remark We develop DLDA independently as a natural extension of LDA to two vocabularies. While writing up the thesis, we discovered [42], which proposes a model with identical structure to DLDA applied to the problem domain of author disambiguation.

We now highlight the following differences between the methods in this chapter and that in [42]. Firstly, in this thesis, the inference techniques we develop are extensions to the Variational Bayes and Online Variational Bayes algorithms for LDA. Secondly, we address a different style of the classification problem: In this thesis, we detect the subject areas of a document, wherein [42], the objective is to disambiguate between a pair of authors. We briefly discuss the outcomes of our experiments compare with the results of [42] in Section 5.4.3.

5.1 Dual Latent Dirichlet Allocation

The Dual Latent Dirichlet Allocation model generalises LDA to observe data over two separate vocabularies. In the context of mathematical document modelling, DLDA assumes that a mathematical document comprising of both words and symbols can be represented by a mixture of latent topics. Moreover, the words and symbols each attribute a latent topic. Unlike LDA however, a topic in the DLDA model is characterised by *two* separate Categorical distributions: a distribution of words and a distribution of symbols.

DLDA assumes a similar generative process to LDA. In particular, DLDA assumes the same generative process of Categorically sampling a collection of words from topics according to the topic mixture of a document, and further Categorically sampling a collection of symbols in the same way.

Formally, this model assumes that the number of latent topics K and the size of the word and symbol vocabularies V and V^s respectively are known and fixed. Furthermore, the numbers of words and symbols in a document N_d and N_d^s respectively are each Poisson distributed with parameters ξ and ξ^s respectively.

Finally, the word and symbol topics are Dirichlet distributed with smoothing parameters $\boldsymbol{\eta}$ and $\boldsymbol{\eta}^s$ respectively. There are separate smoothing parameters on the topics since the vocabularies may be of different sizes.

We note that the DLDA model is not identical to the LDA model over concatenated vocabularies (a single vocabulary spanning words and symbols). Indeed, the DLDA model allows for the numbers of words N_d and symbols N_d^s to belong to different distributions. Under LDA over concatenated vocabularies, the model may only observe the total number of words and symbols combined appearing in a document. This difference manifests in the Variational Bayes algorithms via summations over the two vocabularies. Here the Poisson assumptions are still not critical to anything that follows in the Chapter.

Given the Dirichlet prior $\boldsymbol{\alpha}$ on the topic mixtures, word topics $\boldsymbol{\beta}$, and symbol topics $\boldsymbol{\beta}^s$, DLDA assumes the generative process of a document $(\mathbf{w}_d, \mathbf{s}_d)$ outlined in Algorithm 5.

Algorithm 5 Generative process for a document Under DLDA

Sample topic mixture $\boldsymbol{\theta}_d \sim \text{Dir}(\boldsymbol{\alpha})$.
Sample number of words $N_d \sim \text{Poisson}(\xi)$.
Sample number of symbols $N_d^s \sim \text{Poisson}(\xi^s)$. ▷ New Step
for each of the N_d words **do**
    Sample word topic index $j = z_{dn} \sim \text{Cat}(\boldsymbol{\theta}_d)$.
    Sample word index $\mathbf{w}_{dn} \sim \text{Cat}(\boldsymbol{\beta}_j)$.
end for
for each of the N_d^s words **do** ▷ New Step
    Sample symbol topic index $j = z_{dn}^s \sim \text{Cat}(\boldsymbol{\theta}_d)$. ▷ New Step
    Sample symbol index $\mathbf{s}_{dn} \sim \text{Cat}(\boldsymbol{\beta}_j^s)$. ▷ New Step
end for ▷ New Step

Algorithm 5 makes clear the new steps DLDA introduces to the LDA generative process. In particular, the main difference is that DLDA now assumes a K symbol topics $\boldsymbol{\beta}_j^s$ and for each observation, assumes N^s symbols with corresponding symbol topic indices. We use the same notation as LDA and use a superscript s to denote the symbol specific random variables.

Figure 5-1 shows an example of the topic mixtures for three documents and three word and symbol topics under the DLDA model.

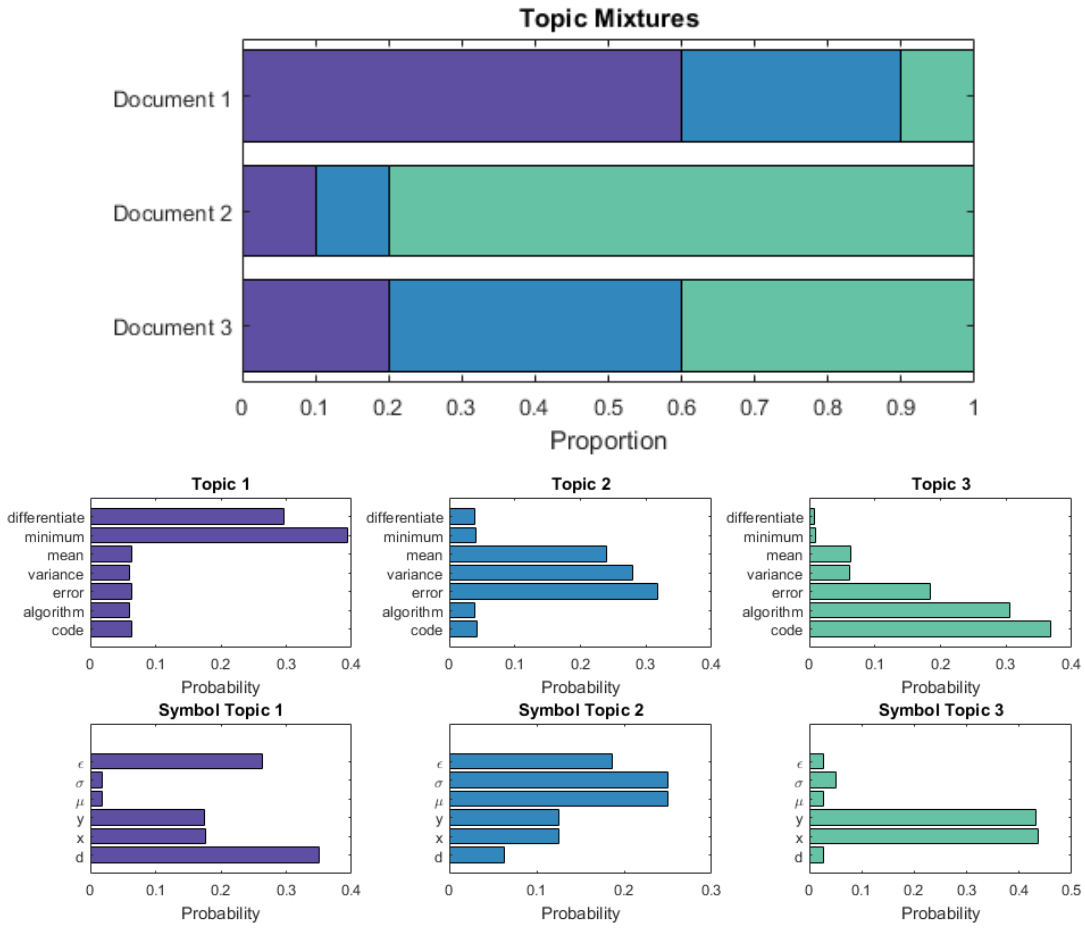

Figure 5-1: Example topic mixtures and corresponding word and symbol topics under DLDA.

We now present the joint distribution of a document and latent variables under DLDA. We first recall the equivalent joint distribution under LDA given by Equation (4.6):

$$p_{\text{LDA}}(\mathcal{H}, \mathbf{w}_d|\boldsymbol{\Theta}) = p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \prod_{j=1}^{K} p(\boldsymbol{\beta}_j|\boldsymbol{\eta}) \prod_{n=1}^{N_d} p(\mathbf{z}_{dn}|\boldsymbol{\theta}_d)\, p(\mathbf{w}_{dn}|\mathbf{z}_{dn}, \boldsymbol{\beta})$$

Naturally, due to the common assumptions of DLDA and LDA, these factors all appear in the the joint distribution of a document and the latent variables given the model parameters under DLDA. The joint distribution of a document $(\mathbf{w}_d, \mathbf{s}_d)$ and the latent variables under DLDA in terms of p_{LDA} and the symbol specific random variables of the model is given by

$$p(\mathcal{H}, \mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta}) = p_{\text{LDA}}(\mathcal{H}, \mathbf{w}_d|\boldsymbol{\Theta}) \prod_{j=1}^{K} p(\boldsymbol{\beta}_j^s|\boldsymbol{\eta}^s) \prod_{n=1}^{N_d^s} p(\mathbf{z}_{dn}^s|\boldsymbol{\theta}_d) p(\mathbf{s}_{dn}|\mathbf{z}_{dn}^s, \boldsymbol{\beta}^s) \quad (5.1)$$

where $\mathcal{H}$ denotes the set of latent variables, $\boldsymbol{\Theta}$ denotes the set of model parameters. The remaining symbol specific factors $p(\boldsymbol{\beta}_j^s|\boldsymbol{\eta}^s)$ and $p(\mathbf{z}_{dn}^s|\boldsymbol{\theta}_d)$ are given by the probability density functions of the Dirichlet distribution and Categorical distribution respectively, and the factor $p(\mathbf{s}_{dn}|\mathbf{z}_{dn}^s, \boldsymbol{\beta}^s)$ is characterised by the probability density function of the Categorical distribution conditioned on z_{dn}^s given by $p(\mathbf{s}_{dn}|\boldsymbol{\beta}_{z_{dn}^s}^s)$.

Figure 5-2 describes the DLDA model as a probabilistic graphical model and makes clear the conditional independence of the word and symbol topic indices given the topic mixture. Furthermore, the similarities to LDA made clear when comparing to the probabilistic graphical model of LDA given in Figure 5-3; removing the symbol specific nodes yields the single vocabulary LDA graphical model.

5.2 Inference

We now outline inference for Dual Latent Dirichlet Allocation. Inference for DLDA is very similar to LDA. Due to the similarities of the structure of the two models, some of the components of the inference procedures are, in fact, identical. In this section, we first introduce a Batch Variational Bayes algorithm which we

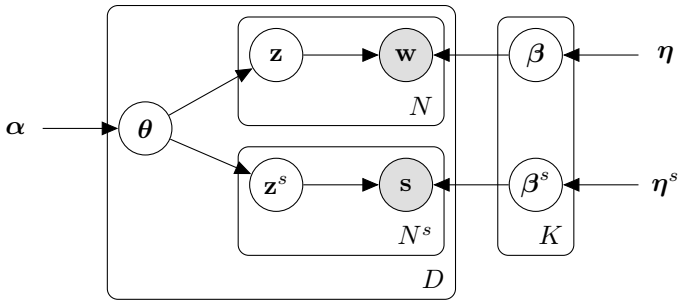

Figure 5-2: Graphical model representation of DLDA

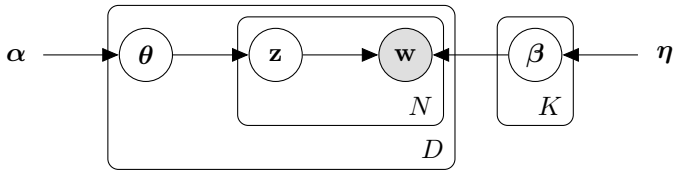

Figure 5-3: Graphical model representation of LDA

then extend to Online Variational Bayes.

We now solve for latent variables in the DLDA model; given the model parameters, we wish to determine the values of the latent variables which maximise the likelihood of a document $(\mathbf{w}_d, \mathbf{s}_d)$ via the posterior distribution

$$p(\mathcal{H}|\mathbf{w}_d, \mathbf{s}_d, \boldsymbol{\Theta}) = \frac{p(\mathcal{H}, \mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta})}{p(\mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta})} \tag{5.2}$$

Similar to LDA, this posterior distribution is intractable to compute in general due to the coupling between the topic mixture and the topics [32], and we instead employ Variational Bayes to approximate this posterior.

5.2.1 Variational Inference

Variational Bayes for DLDA is almost identical to Variational Bayes for LDA. Recall that the strategy for variational inference is to obtain the tightest possible lower bound on the log-likelihood [33] by optimising this lower bound over a set of free variational parameters.

We construct this lower bound using the same strategy as in LDA described in Section 4.3.1. We consider a simpler version of the graphical model in Figure

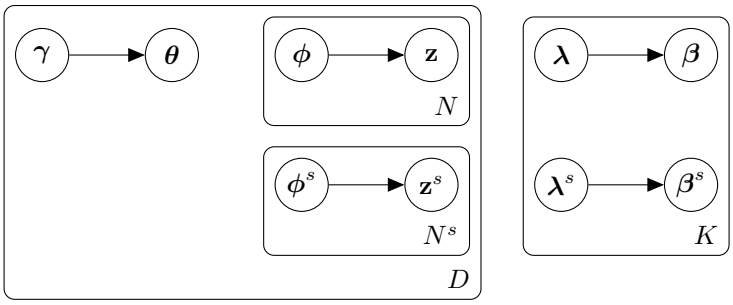

Figure 5-4: Graphical model representation of the variational distribution of DLDA

5-2 with problematic nodes and edges removed and augment this model with a set of free variational parameters: we remove all edges, observed variables, and model parameters; we augment the model with the variational parameters $\boldsymbol{\gamma}$, $\boldsymbol{\phi}$ and $\boldsymbol{\lambda}$ in the same fashion as LDA, and finally, the equivalent symbol specific variational parameter $\boldsymbol{\phi}^s$ and $\boldsymbol{\lambda}^s$. Formally, we consider the variational distribution characterised by

$$q(\mathcal{H}|\mathcal{F}) = \prod_{j=1}^{K} q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j) q(\boldsymbol{\beta}_j^s|\boldsymbol{\lambda}_j^s) \prod_{d=1}^{D} q_d(\mathcal{H}|\mathcal{F}) \tag{5.3}$$

where $\mathcal{F}$ denotes the set of free variational parameters, and q_d characterises the variational distribution of the dth document given by

$$q_d(\mathcal{H}|\mathcal{F}) = q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \prod_{n=1}^{N_d} q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn}) \prod_{n=1}^{N_d^s} q(\mathbf{z}_{dn}^s|\boldsymbol{\phi}_{dn}^s) \tag{5.4}$$

where the symbol specific factors $q(\boldsymbol{\beta}_j^s|\boldsymbol{\lambda}_j^s)$ and $q(\mathbf{z}_{dn}^s|\boldsymbol{\phi}_{dn}^s)$ are given by the probability density functions of the Dirichlet and Categorical distributions respectively, and the remaining factors are given by the same probability density functions as in the LDA variational distribution. Figure 5-4 describes the full variational distribution of DLDA as a probabilistic graphical model.

We obtain a lower bound of the log-likelihood using Jensen's inequality via the same method employed when deriving the lower bound for the log-likelihood

of a document under LDA and arrive at the inequality

$$\log p(\mathbf{w}, \mathbf{s}|\boldsymbol{\Theta}) \geq \mathbb{E}_q[\log p(\mathcal{H}, \mathbf{w}, \mathbf{s}|\boldsymbol{\Theta})] - \mathbb{E}_q[\log q(\mathcal{H}|\mathcal{F})] \tag{5.5}$$

We let $\mathcal{L}$ denote the right-hand side of the above inequality as a function of the free variational parameters $\mathcal{F}$ given the model parameters $\boldsymbol{\Theta}$ and call this the *Evidence Lower Bound*. We rewrite $\mathcal{L}$ using the factorisations of p and q and arrive at

$$\begin{aligned}\mathcal{L}(\mathcal{F}; \boldsymbol{\Theta}) = \sum_{d=1}^{D} L_d(\mathcal{F}; \boldsymbol{\Theta}) + \mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] + \mathbb{E}_q[\log p(\boldsymbol{\beta}^s|\boldsymbol{\eta}^s)] \\ - \mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] - \mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda}^s)]\end{aligned} \tag{5.6}$$

where L_d denotes the contribution of the dth document to the Evidence Lower Bound. Again, due to the similarities between DLDA and LDA, we discover that this contribution is the same as the equivalent contribution under LDA, but with the addition of the symbol specific expectations

$$\begin{aligned}L_d(\mathcal{F}; \boldsymbol{\Theta}) = L_d^{\text{LDA}}(\mathcal{F}; \boldsymbol{\Theta}) + \mathbb{E}_q[\log p(\mathbf{z}_d^s|\boldsymbol{\theta}_d)] + \mathbb{E}_q[\log p(\mathbf{s}_d|\mathbf{z}^s, \boldsymbol{\beta}^s)] \\ - \mathbb{E}_q[\log q(\mathbf{z}_d^s|\boldsymbol{\phi}_d^s)]\end{aligned} \tag{5.7}$$

where L_d^{LDA} corresponds to the equivalent contribution of the dth document under LDA given by Equation (4.11).

We now have a lower bound on the log-likelihood of a corpus given an arbitrary variational distribution q. Similar to LDA, we wish to minimise the difference between the left-hand side and the right-hand side of Equation (5.5) where this difference is indeed the KL divergence between the posterior probability $q(\mathcal{H}|\mathcal{F})$ and the true posterior probability $p(\mathcal{H}|\mathbf{w}, \mathbf{s}, \boldsymbol{\Theta})$. We express the log-likelihood as a function of the free variational parameters via the Evidence Lower Bound and this KL divergence to give

$$\log p(\mathbf{w}, \mathbf{s}|\boldsymbol{\Theta}) = \mathcal{L}(\mathcal{F}; \boldsymbol{\Theta}) + \mathrm{D}(q(\mathcal{H}|\mathcal{F}) \,\|\, p(\mathcal{H}|\mathbf{w}, \mathbf{s}, \boldsymbol{\Theta}))$$

Here we see that minimising the KL divergence between the variational posterior probability and the true posterior probability is equivalent to maximising $\mathcal{L}$ via the free variational parameters. We now outline the steps for maximising

the Evidence Lower Bound.

We express each of the expectations in Equations (5.6) and (5.7) as their expanded forms outlined in Appendix B. Furthermore, we discover that the expectations over the topics $\boldsymbol{\beta}$, and those residing in L_d^{LDA} have identical expansions to the LDA case. The remaining expectations in Equation (5.6) expand as follows.

Appendices B.2 and B.3 show the expectations over the symbol topics $\boldsymbol{\beta}^s$ expand to

$$\mathbb{E}_q[\log q(\boldsymbol{\beta}^s|\boldsymbol{\lambda}^s)] = \sum_{j=1}^{K}\left(\log\Gamma\left(\sum_v \lambda_{jv}^s\right) + \sum_{v=1}^{V^s}\left((\lambda_{jv}^s - 1)\mathbb{E}_q\left[\log\beta_{jv}^s|\boldsymbol{\lambda}_j^s\right] - \log\Gamma\left(\lambda_{jv}^s\right)\right)\right)$$

$$\mathbb{E}_q[\log p(\boldsymbol{\beta}^s|\boldsymbol{\eta}^s)] = K\left(\log\Gamma\left(\sum_v \eta_v^s\right) - \sum_{v=1}^{V^s}\log\Gamma(\eta_v^s)\right) + \sum_{v=1}^{V^s}(\eta_v^s - 1)\sum_{j=1}^{K}\mathbb{E}_q\left[\log\beta_{jv}^s|\boldsymbol{\lambda}_j^s\right]$$

where we emphasise the dependency on $\boldsymbol{\lambda}^s$ in the inner expectations. Appendices B.4 and B.5 show the expectations over the symbol topic indices expand and rearrange to give the summations

$$\mathbb{E}_q[\log p(\mathbf{z}_d^s|\boldsymbol{\theta}_d)] = \sum_{n=1}^{N_d^s}\sum_{j=1}^{K}\phi_{dnj}^s\mathbb{E}_q[\log\theta_{dj}|\boldsymbol{\gamma}_d]$$

$$\mathbb{E}_q[\log q(\mathbf{z}_d^s|\boldsymbol{\phi}_d^s)] = \sum_{n=1}^{N_d^s}\sum_{j=1}^{K}\phi_{dnj}^s\log\phi_{dnj}^s$$

where we emphasis the dependencies on $\boldsymbol{\gamma}_d$ in the inner expectations. Finally, Appendix B.6 shows the expectations over the symbols $\mathbf{s}_d$ expand and rearranges to give the summation

$$\mathbb{E}_q[\log p(\mathbf{s}_d|\boldsymbol{\beta}^s, \mathbf{z}_d^s)] = \sum_{n=1}^{N_d^s}\sum_{j=1}^{K}\phi_{dnj}^s\mathbb{E}_q\left[\log\beta_{js_{dn}}^s|\boldsymbol{\lambda}_j^s\right]$$

where we emphasise the dependency on $\boldsymbol{\lambda}^s$ in the inner expectations.

By plugging in these expectations into Equation (5.6), we obtain the lower bound $\mathcal{L}$ as a function of the model parameters and the free variational parameters. We now outline the process of maximising this lower bound via the variational parameters.

5.2.1.1 Document Level Updates

In this section, we describe the methods of maximising $\mathcal{L}$ with respect to each of the document level variational parameters $\boldsymbol{\phi}_d$, $\boldsymbol{\phi}_d^s$ and $\boldsymbol{\gamma}_d$.

Variational Categorical Parameters The strategy for finding the maximising values of ϕ_{dnj} under DLDA is identical to finding the maximising values of ϕ_{dnj} under LDA as described in Section 4.3.1. Furthermore, the maximising values for ϕ_{dnj} are identical, this is due to the DLDA model not introducing any more terms to $\mathcal{L}$ which are dependent on ϕ_{dnj}.

Following the same strategy for determining the maximising values of ϕ_{dnj} under LDA, we discover that the maximising values are of the same form as ϕ_{dnj} but with the word specific and symbol specific variables interchanged. In particular, the maximising value of ϕ_{dnj}^s is given by

$$\phi_{dnj}^s \propto \exp\left\{\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q\left[\log \beta_{js_{dn}}^s|\boldsymbol{\lambda}_j^s\right]\right\}$$

where $\boldsymbol{\phi}_{dn}^s$ is normalised to sum to one.

Variational Dirichlet Parameters We now maximise the Evidence Lower Bound via γ_{dj}; the jth component of the Dirichlet parameter on the topic mixtures of the dth document. The maximising value of γ_{dj} is *almost* identical to the LDA equivalent in Section 4.3.1. There are a few additional terms in $\mathcal{L}$ which contain γ_{dj} due to the dependency of $\mathbf{z}_d^s$ on $\boldsymbol{\theta}_d$. Retaining only the terms of $\mathcal{L}$ containing γ_{dj} we have

$$\mathcal{L}_{[\gamma_{dj}]} = \sum_{j'=1}^{K} \mathbb{E}_q[\log \theta_{dj'}|\boldsymbol{\gamma}_d]\left(\alpha_{j'} + \sum_{n=1}^{N_d}\phi_{dnj} + \underbrace{\sum_{n=1}^{N_d^s}\phi_{dnj}^s}_{\text{New terms}} - \gamma_{dj'}\right)$$
$$- \log\Gamma\left(\sum_{j'}\gamma_{dj'}\right) + \log\Gamma(\gamma_j)$$

where we highlight the new terms introduced by this model. Taking partial derivatives with respect to γ_{dj} yields

$$\frac{\partial \mathcal{L}_{[\gamma_{dj}]}}{\partial \gamma_{dj}} = \Psi'(\gamma_{dj})\left(\alpha_j + \sum_{n=1}^{N_d}\phi_{dnj} + \sum_{n=1}^{N_d^s}\phi^s_{dnj} - \gamma_{dj}\right) - \Psi'\left(\sum_{j'}\gamma_{dj'}\right)\sum_{j'=1}^{K}\left(\alpha_{j'} + \sum_{n=1}^{N_d}\phi_{dnj'} + \sum_{n=1}^{N_d^s}\phi^s_{dnj'} - \gamma_{dj'}\right)$$

Finally, setting this derivative to zero yields a maximising value of γ_{dj} at

$$\gamma_{dj} = \alpha_j + \sum_{n=1}^{N_d}\phi_{dnj} + \underbrace{\sum_{n=1}^{N_d^s}\phi^s_{dnj}}_{\text{New terms}}$$

where again we highlight the new terms introduced by this model compared to the LDA equivalent.

We now have the update rules for the variational parameters which we require for the document level variational updates. Since the update rules for the $\boldsymbol{\phi}_d$ and $\boldsymbol{\phi}^s_d$ are dependent on $\boldsymbol{\gamma}_d$ and vice versa, full variational inference requires alternating between these update rules until convergence.

Notation

As evident in the update rules for ϕ_{dnj} and ϕ^s_{dnj}, due to the symmetrical nature of the models introduced in this thesis, we may encounter pieces of mathematics which are identical up to the exchange of the word specific variables with the corresponding symbol specific variables. If this is the case, for simplicity we may use an asterisk placeholder to denote this exchangeability. For example, instead of presenting both of the formulae

$$\phi_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\}$$
$$\phi^s_{dnj} \propto \exp\{\mathbb{E}_q\left[\log \theta^s_{dj}|\boldsymbol{\gamma}^s_d\right] + \mathbb{E}_q\left[\log \beta^s_{js_{dn}}|\boldsymbol{\lambda}^s_j\right]\}$$

which differ only by the use of superscript s, we may instead present the following single formula with a superscript asterisk

$$\phi^*_{dnj} \propto \exp\left\{\mathbb{E}_q\left[\log \theta^*_{dj} | \boldsymbol{\gamma}^*_d\right] + \mathbb{E}_q\left[\log \beta^*_{j*_{dn}} | \boldsymbol{\lambda}^*_j\right]\right\}$$

We summarise the document level variational inference procedure for Dual Latent Dirichlet Allocation in Algorithm 6.

Algorithm 6 Document level variational inference for Dual Latent Dirichlet Allocation

Initialise γ_d randomly

repeat

Set $\phi_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj} | \boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}} | \boldsymbol{\lambda}_j]\}$

Set $\phi^s_{dnj} \propto \exp\left\{\mathbb{E}_q[\log \theta_{dj} | \boldsymbol{\gamma}_d] + \mathbb{E}_q\left[\log \beta^s_{js_{dn}} | \boldsymbol{\lambda}^s_j\right]\right\}$

Set $\boldsymbol{\gamma}_{dj} = \alpha_j + \sum_n \phi_{dnj} + \sum_n \phi^s_{dnj}$

until Convergence of $\boldsymbol{\gamma}_d$

Similar to LDA, the optimisation is conducted for a single fixed document $(\mathbf{w}_d, \mathbf{s}_d)$. Therefore, we can use the procedure that yields the optimised values of $\boldsymbol{\phi}_d$, to approximate the topic mixture representation of a given document. In particular, we evaluate the expected value of the normalised frequency count of the topic indices $\mathbf{z}_{dn}$ and $\mathbf{z}^s_{dn}$ under the variational distribution q:

$$\bar{\boldsymbol{\phi}}_d = \mathbb{E}_q\left[\frac{1}{N_d + N^s_d}\left(\sum_{n=1}^{N_d} \mathbf{z}_{dn} + \sum_{n=1}^{N^s_d} \mathbf{z}^s_{dn}\right)\right] = \frac{1}{N_d + N^s_d}\left(\sum_{n=1}^{N_d} \boldsymbol{\phi}_{dn} + \sum_{n=1}^{N^s_d} \boldsymbol{\phi}^s_{dn}\right)$$

where we use the fact that under the variational distribution q, the values of $\mathbf{z}_{dn}$ and $\mathbf{z}^s_{dn}$ are Categorically distributed with parameters $\boldsymbol{\phi}_{dn}$ and $\boldsymbol{\phi}^s_{dn}$ respectively.

5.2.1.2 Corpus Level Updates

We now have a variational inference procedure for approximating the document level variational parameters. We now derive the full Variational Bayes method for approximating the variational parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$, and the Dirichlet prior $\boldsymbol{\alpha}$ which maximise the marginal log-likelihood of the corpus.

We have already shown that there is a tractable lower bound on the log-likelihood, and we can further maximise this lower bound via the model parameter $\boldsymbol{\alpha}$ and the variational parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$. Therefore, we can derive a full variational EM procedure that yields the optimised values for $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$.

We optimise for $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$ using the same method to optimise $\boldsymbol{\lambda}$ under LDA and discover that the maximising value for $\boldsymbol{\lambda}$ is identical, and furthermore, the maximising value for $\boldsymbol{\lambda}^s$ is the same but with the word specific and symbol specific variables exchanged. In particular, the maximising values of λ^s_{jv} are given by

$$\lambda^s_{jv} = \eta^s_v + \sum_{d=1}^{D} \sum_{n=1}^{N^s_d} s_{dnv} \phi^s_{dnj}$$

Finally, since the DLDA model does not introduce any new variables dependent on $\boldsymbol{\alpha}$ to the LDA model, we again use the Newton-Raphson method described in [36] to find the maximising value of $\boldsymbol{\alpha}$.

We now have the required document and corpus level updates required for full Variational Bayes for DLDA. The outline of the full variational EM procedure is as follows

- E-step: For each document, find the optimised values of the variational parameters $\boldsymbol{\gamma}_d$, $\boldsymbol{\phi}_d$, and $\boldsymbol{\phi}^s_d$ using the document level updates.
- M-step: Maximise the resulting lower bound on the log-likelihood via the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\lambda}^s$.

We summarise the full variational inference procedure on a corpus of D mathematical documents in Algorithm 7. We apply the document level variational updates for each document in the corpus, and update the corpus level parameters after each pass of the corpus. We make clear the similarity to Batch Variational Bayes for LDA by highlighting the additional and modified steps.

Algorithm 7 Batch Variational Bayes for Dual Latent Dirichlet Allocation

Initialise $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^s$ randomly ▷ Additional Step

repeat

  for $d = 1, \ldots, D$ **do**

    Initialise γ_d randomly

    repeat

      Set $\phi_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\}$

      Set $\phi^s_{dnj} \propto \exp\{\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\gamma}_d] + \mathbb{E}_q[\log \beta^s_{js_{dn}}|\boldsymbol{\lambda}^s_j]\}$ ▷ Additional Step

      Set $\boldsymbol{\gamma}_{dj} = \alpha_j + \sum_n \phi_{dnj} + \sum_n \phi^s_{dnj}$ ▷ Modified Step

    until Convergence of $\boldsymbol{\gamma}_d$

  end for

  Set $\lambda_{jv} = \eta_v + \sum_d \sum_n w_{dnv}\phi_{dnj}$

  Set $\lambda^s_{jv} = \eta^s_v + \sum_d \sum_n s_{dnv}\phi^s_{dnj}$ ▷ Additional Step

  Update $\boldsymbol{\alpha}$ according to [36]

until Convergence of $\mathcal{L}$

DLDA is not identical to LDA over concatenated vocabularies. DLDA assumes that the distributions of the numbers of words and symbols appearing in a document may be different which manifests in the summations over $\boldsymbol{\lambda}^*_j$ when evaluating the expectation of the log topics.

5.2.2 Online Variational Inference

In Algorithm 7, we encounter the same issues as in Batch Variational Bayes for LDA regarding the efficiency of the algorithm. In particular, this batch algorithm requires many passes of the corpus and, in turn, has large memory requirements.

We address this problem in the same way: we develop Online Variational Bayes for DLDA which requires only a single pass of the corpus. In particular, we now update the Variational Dirichlet parameters on the topics $\boldsymbol{\lambda}^*$ at each iteration using a weighted average of their previous values and the optimal value according to the current values of $\boldsymbol{\phi}^*_d$ via the weighting parameter ρ_d. The definition of the weighting parameter ρ_d and method for updating the Dirichlet prior $\boldsymbol{\alpha}$ is identical to Online Variational Bayes for LDA.

We describe Online Variational Bayes for Dual Latent Dirichlet Allocation in

Algorithm 8.

Algorithm 8 Online Variational Bayes for Dual Latent Dirichlet Allocation

Define $\rho_d := (\tau_0 + d)^{-\kappa}$
Initialise $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^s$ randomly
for $d = 1, 2, \ldots$ **do**
  Initialise γ_d randomly
  repeat
    Set $\phi_{dnj} \propto \exp\left\{\Psi(\gamma_{dj}) + \Psi(\lambda_{jw_{dn}}) - \Psi\left(\sum_v \lambda_{jv}\right)\right\}$
    Set $\phi^s_{dnj} \propto \exp\left\{\Psi(\gamma_{dj}) + \Psi\left(\lambda^s_{js_{dn}}\right) - \Psi\left(\sum_v \lambda^s_{jv}\right)\right\}$
    Set $\gamma_{dj} = \alpha_j + \sum_n \phi_{dnj} + \sum_n \phi^s_{dnj}$
  until Convergence of $\boldsymbol{\gamma}_d$
  Set $\tilde{\lambda}_{jv} = \eta_v + D \sum_n w_{dnv} \phi_{dnj}$
  Set $\tilde{\lambda}^s_{jv} = \eta^s_v + D \sum_n s_{dnv} \phi^s_{dnj}$
  Set $\boldsymbol{\lambda} = (1 - \rho_d)\boldsymbol{\lambda} - \rho_d \tilde{\boldsymbol{\lambda}}$
  Set $\boldsymbol{\lambda}^s = (1 - \rho_d)\boldsymbol{\lambda}^s - \rho_d \tilde{\boldsymbol{\lambda}}^s$
  Update $\boldsymbol{\alpha}$ according to [6]
end for

5.3 Document Classification

We now outline mathematical document classification via Dual Latent Dirichlet Allocation. The framework is remarkably similar to document classification via LDA as described in Section 4.4. In particular, we describe a semi-supervised classifier trained on a partially labelled collection of mathematical documents via their topic mixture representations.

5.3.1 Framework

For completeness, we outline the complete framework of document classification via Dual Latent Dirichlet Allocation. As before, we break down the process into two layers: the unsupervised layer (the document modelling step), and the supervised layer (the supervised training step).

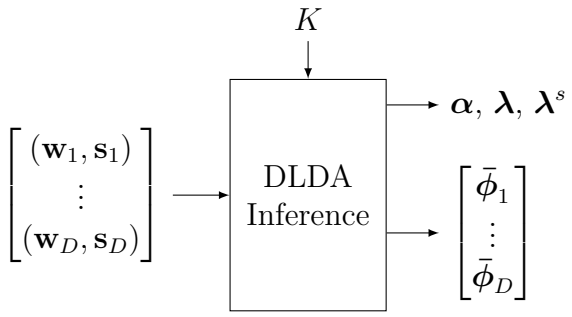

Figure 5-5: The Unsupervised Layer using DLDA

Document Modelling

Given a partially labelled corpus and a choice of K, we use the DLDA variational inference methods to approximate the Dirichlet prior $\boldsymbol{\alpha}$, the Variational Dirichlet parameters $\boldsymbol{\lambda}^*$ on the topics, and the topic mixture representations of the documents $\bar{\boldsymbol{\phi}}_d$. Figure 5-5 shows a diagrammatic illustration of this layer.

Compared with classification via LDA, the unsupervised layer for classification via DLDA requires the symbol data from the documents and outputs one extra parameter: the variational parameter $\boldsymbol{\lambda}^s$ on the symbol topics.

Supervised Training

The supervised layer is identical to the one we describe in classification via LDA framework in Section 4.4. Using the *labelled* topic mixture representations obtained from the unsupervised layer as feature vectors, we train a supervised classifier. Again, we may use any discriminative supervised classification methods. We continue to use nearest neighbour methods for supervised classification which we describe in Appendix C.2.

Classification

We now have all the tools necessary for mathematical document classification via DLDA. The unsupervised layer provides the required parameters to obtain the topic mixture representation $\bar{\boldsymbol{\phi}}$ of an unseen document. The supervised layer provides the document classifier f which will output the predicted set of labels $\mathbf{c}$ for this document. Figure 5-6 outlines the classification process of a previously

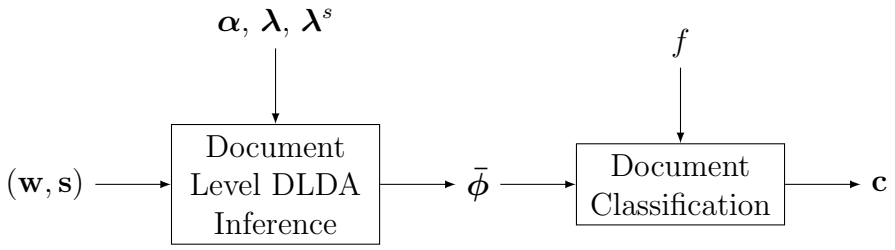

Figure 5-6: Classification process via DLDA

unseen document.

To summarise, DLDA provides another fast filtering algorithm for feature selection for *mathematical* document classification. In particular, over LDA, DLDA provides a stronger topic mixture representation since this representation also utilises symbol information while maintaining the same significant dimension reduction. We may instead consider mathematical documents containing possibly thousands of word and symbol features as a mixture of a significantly smaller number of topics.

5.4 Experimental Results

We now evaluate the performance of mathematical document classification via DLDA. For our experiments, we continue to use the same experimental set-up as before. In particular, we observe the effect that the choice of the number of topics K has on classification performance, we optimise the best performing classifier to obtain the optimised Labelling F-Score by adjusting the decision thresholds on the supervised classifiers, and finally, we look closely at the outputs of the best performing classifiers and investigate the per-class confusion.

5.4.1 Preliminary Experiments

To get an idea of how classification via DLDA behaves, we run our experiments for a selection of values of K and observe the effect it has on classification performance. For our preliminary experiments, we use the same set-up as our preliminary experiments for LDA in Section 4.5. We perform Dual Latent Dirichlet Allocation on our training data using the values of $K \in \{10, 25, 50, 100, 200, 250\}$.

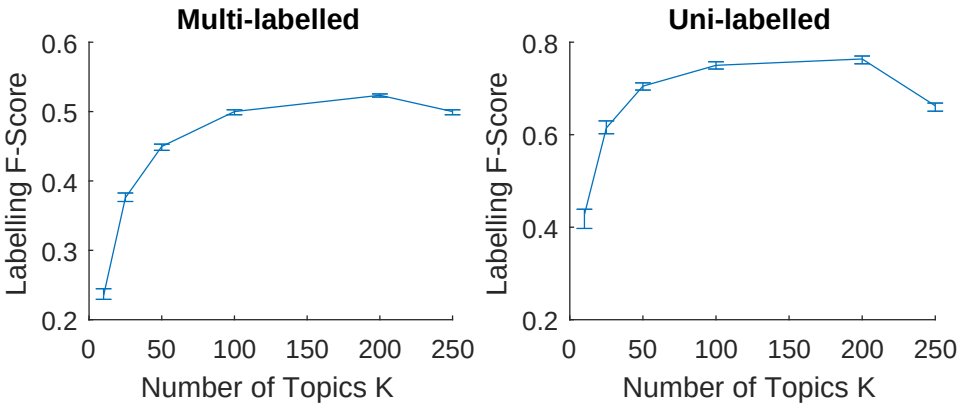

Figure 5-7: Classification performance via DLDA

For the supervised layer, we train an ensemble of Nearest Neighbour Classifiers using five nearest neighbours, inverse distance weighting and the χ^2 distance metric.

Effect of K

Figure 5-7 shows the effect that the choice of values $K \in \{10, 25, 50, 100, 200, 250\}$ has on classification performance via DLDA on the multi-labelled and uni-labelled datasets. We repeat each experiment sixteen times on different random test and train partitions of the data.

We see similarly shaped curves when compared to classification via LDA. In particular, we see in increase in classification performance as the choice of K increases until the $K = 200$ mark before we see the classification performance decrease.

Precision/Recall Trade-off

Recall that in the LDA setting; we notice an imbalance between the precision and recall of the classifiers. We discover that this is also the case for our DLDA classifier. In particular, we observe micro-averaged precision and recall of 77.94% and 40.00% respectively on the uni-labelled dataset and 86.59% and 76.37% respectively on the multi-labelled dataset. As before, we repeat our experiments on the $K = 200$ case with the binary classifiers equipped with a range of different

values of decision thresholds and determine that on both datasets, a threshold of 0.3 yields optimal performance which we outline in Table 5.1.

	Multi-labelled	Uni-labelled
K	200	200
Threshold	0.3	0.3
Labelling F-Score		
(median)	59.96%	80.95%
(maximum)	60.39%	82.33%
Micro-averaged F-Score		
(median)	58.44%	81.87%
(maximum)	59.00%	83.03%

Table 5.1: Optimised Classification Performance via Online Dual Latent Dirichlet Allocation

We now look closely at one of the experiments from the $K = 200$ case at random equipped with this optimal decision threshold.

5.4.2 Confusion

In this section, we investigate the per-class confusion of the classification results and furthermore, briefly compare performance to classification via LDA. Later, in Chapter 8, we will collect the results and compare all models presented in this thesis.

Figure 5-8 shows a confusion matrix highlighting the true per-class true positive rates on the diagonal, false negative rates broken down by false positives over each other class on the off-diagonal, and finally, per-class null classification rates on the final column. Unfortunately, we observe some stronger levels of confusion compared to before. We again see the strongest confusion between related subject areas, but also with higher rates of misclassified instances. In particular, we see the most confusion between the subject areas "Ordinary differential equations" are misclassified as "Partial differential equations". These misclassified instances have a significant effect on the true positive rates which we observe as low as 29.0%.

We now construct the equivalent confusion matrix for multi-labelled classification. Again, due to the size of the confusion matrix, we only present three sections

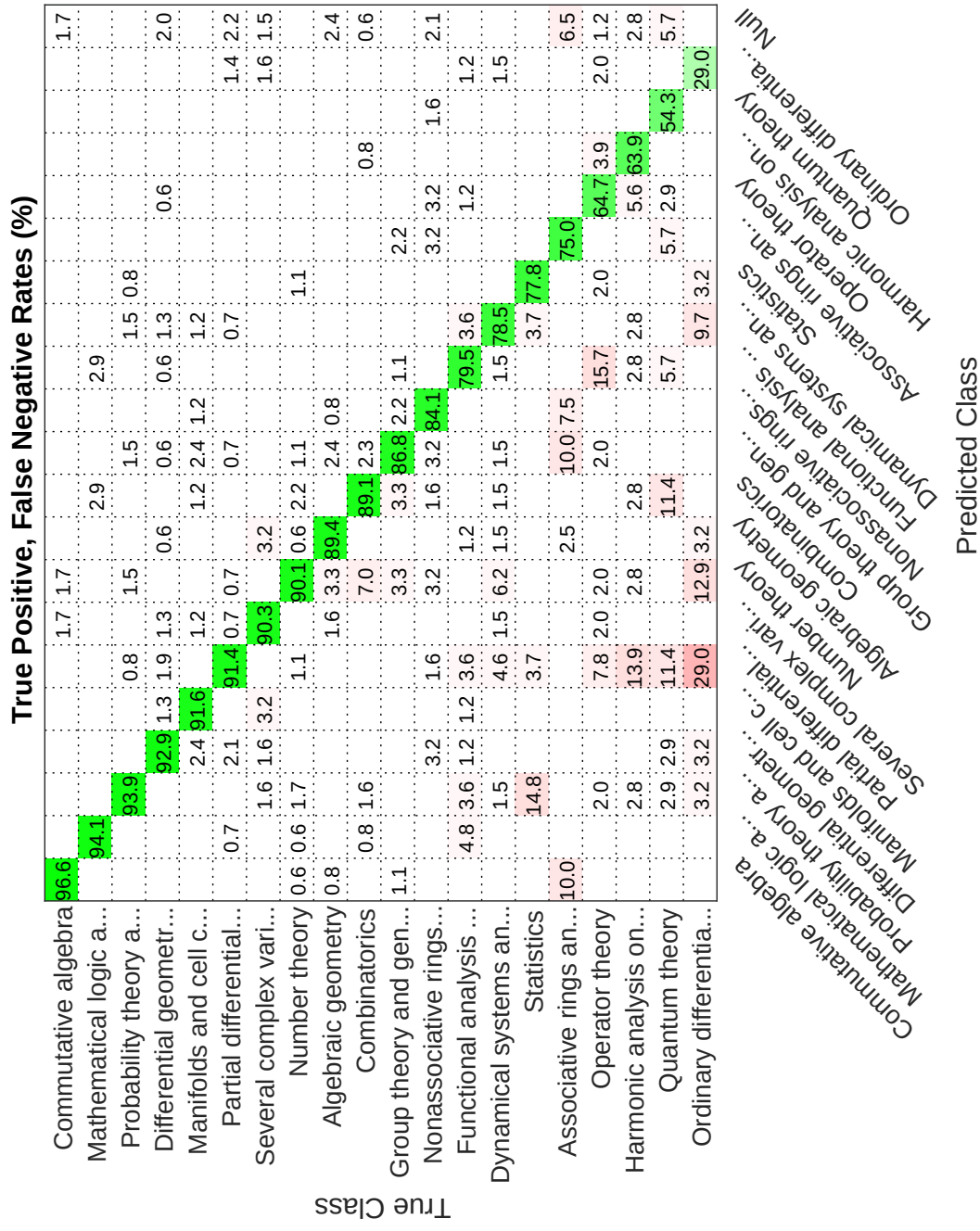

Figure 5-8: Per-class confusion of classification via DLDA on the uni-labelled dataset

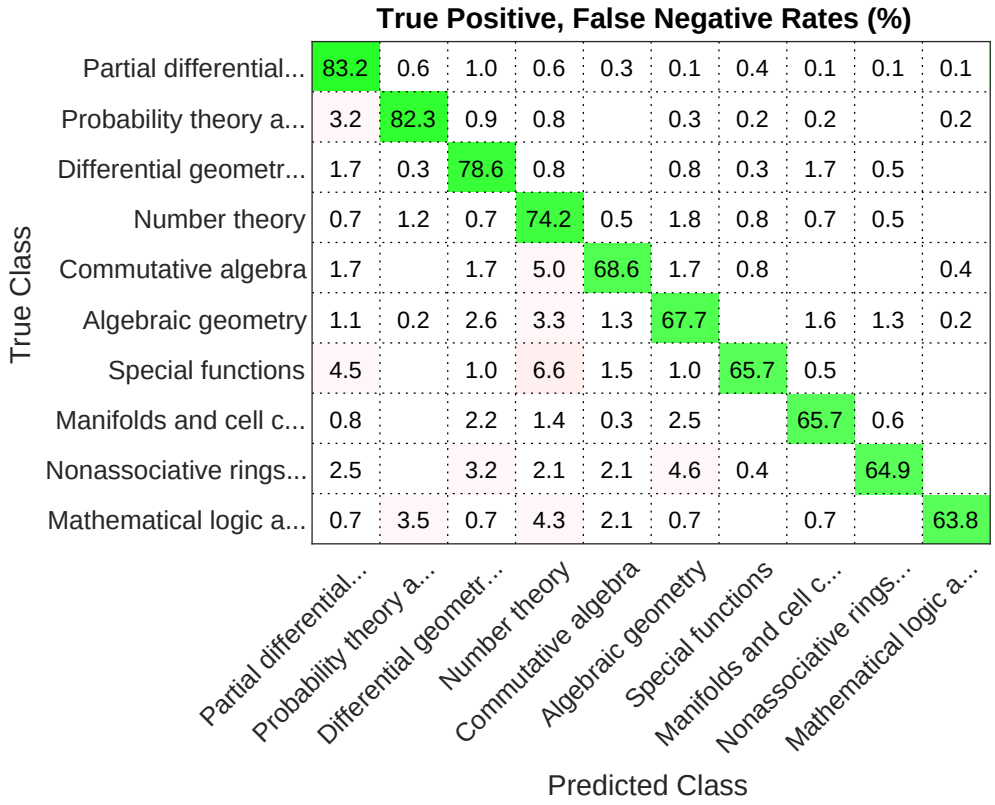

Figure 5-9: Per-class confusion of classification via DLDA on the multi-labelled dataset - Upper-left section.

of the matrix highlighting the areas of strongest and weakest performance. We present the full heat-map of confusion in Appendix D.

Firstly, Figure 5-9 shows the section of the confusion matrix highlighting the best per-class true positive rates; the upper-left section of the confusion matrix. Here we see relatively strong true positive rates amongst the top ten subject areas. We also see a general wash of slight confusion between nearly all pairs of the subject areas here. The true positive and false negative rates in this section of the confusion matrix are very similar in value to the equivalent matrix under LDA.

Figure 5-10 shows the section of the confusion matrix highlighting some of the highest levels of confusion between subject areas; the lower-left section of the confusion matrix. Here we notice some strong levels of confusion, again between related subject areas. In particular, we notice the highest rates of confusion with

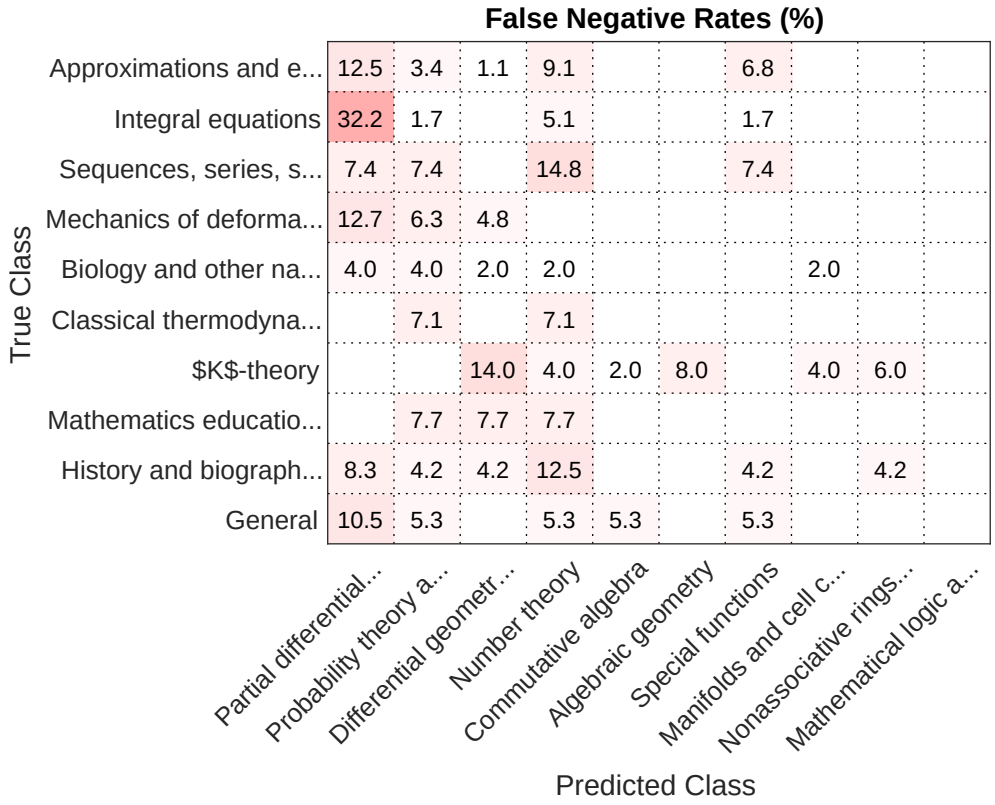

Figure 5-10: Per-class confusion of classification via DLDA on the multi-labelled dataset - Lower-left section.

32.2% of documents positively labelled with "Integral equations" are incorrectly classified as "Partial differential equations".

Finally, Figure 5-11 shows the section of the confusion matrix highlighting some of the worst per-class true positive rate; the lower-right section of the confusion matrix. Similar to LDA, we notice the weakest true positive rates amongst the subject areas close to applied sciences and also, the highest rates of null classifications amongst the more general subject areas such as "General" and "History and biography".

5.4.3 Discussion

Unfortunately, we observe a decline in classification performance here in comparison to classification via LDA. We recall the main assumption of the DLDA

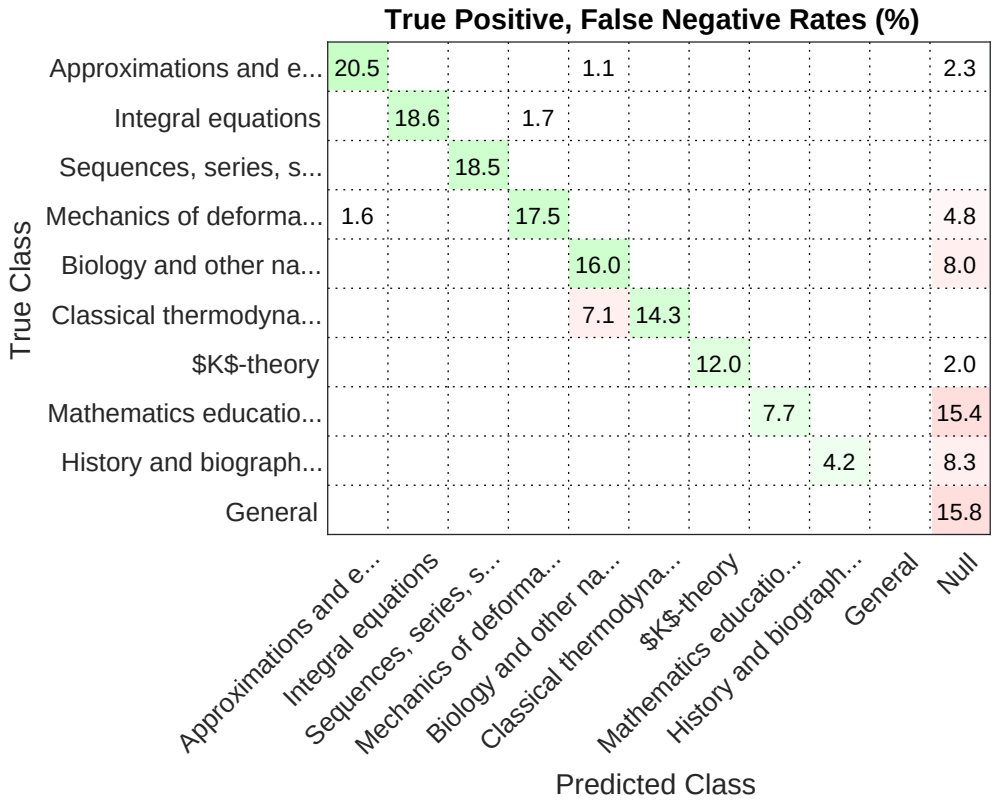

Figure 5-11: Per-class confusion of classification via DLDA on the multi-labelled dataset - Lower-right section.

model: given a mathematical document, the word topic indices and symbol topic indices are distributed according to the documents topic mixture. Under this assumption, the distribution of symbols across topics is identical to the distribution of words across topics. For example, if 50% of the words in a document belong to the same topic, then it must follow that 50% of the symbols must also belong to the same topic. This assumption implies that there must be a one to one correspondence between word topics and symbol topics.

Due to the decline in performance, this leads us to believe that the assumptions made by the DLDA may be incorrect. We may conclude that the collections of word correlations do not behave in the same way as the collections of symbol correlations. In particular, for each word topic, there is not a corresponding symbol topic. That is not to say that meaningful symbol topics do not exist. Indeed, we remind ourselves of the claim in [1]: for each mathematical subject area, the ordered set of most commonly used symbols is unique.

We may have an issue with subject areas which simply "employ" other subject areas. For example, consider documents tagged with "Statistics" and documents that simply use statistical methods such as "Biology and other natural sciences". These documents are likely to use very similar mathematical notation, yet the collections of words will differ greatly. Under DLDA, these situations may force the word topics to be much broader which negatively impacts the discriminative properties of the topic mixture representation.

In the context of author disambiguation in [42], the equivalent of the above observation does not necessarily occur. In particular, the experimental set-up of [42] selects training documents by recursively selecting documents attributed to each author and co-author and by design, there will be strong, distinct correlations between authors (the research groups) and strong correspondences between these research groups and the word topics. In contrast to the mathematical document modelling scenario, these correspondences are unlikely to show any significant overlap. Furthermore, we would not expect such high levels name sharing and name variants of authors compared to the sharing and variants of mathematical symbols hence the strong disambiguation performance in [42].

We conclude that classification via DLDA is not appropriate for mathematical corpora. We now seek a model which observes word correlations as word topics, symbol correlations as symbol topics but allows a document to be characterised by

different mixtures of the two. We achieve this via Dual Pachinko Allocation: our hierarchical generalisation of the DLDA model which discovers word and symbol correlations separately, and further characterises an extra level of correlations as super-topics. Moreover, Dual Pachinko Allocation allows us to represent a document as a mixture of both word topics and symbol topics separately. Before we introduce Dual Pachinko Allocation, we first introduce the single vocabulary Pachinko Allocation model in Chapter 6.

Chapter 6

Pachinko Allocation

In the previous chapter, we introduce Dual Latent Dirichlet Allocation, an extension of the Latent Dirichlet Allocation model which models mathematical documents as mixtures of topics, where each topic is characterised by a distribution of words, and a distribution of symbols. Our experimental results show that classification via DLDA yields worse performance than classification via LDA and in turn, we reject the assumption of DLDA that there is a one-to-one correspondence between word topics and symbol topics.

We now begin to develop a model which allows for word topics and symbols topics to be modelled independently. We impose the assumption that a mathematical document can be represented by a mixture of *two* collections of latent topics: a mixture of word topics and symbol topics. Having these two separate spaces of topics allow for the symbol topics of a document to be distributed differently to the word topics. Moreover, we allow the number of word topics and symbol topics to be different. We achieve this via *Dual Pachinko Allocation*, a hierarchical generalisation of the LDA model, which assumes mathematical documents can be represented as a mixture of latent word topics, symbol topics, and a mixture of latent super-topics which we describe in Chapter 7.

Before introducing Dual Pachinko Allocation, we first introduce the single vocabulary Pachinko Allocation model as described in [5]. *Pachinko Allocation* is a hierarchical generalisation of LDA which discovers arbitrary levels of topic correlations. We focus on a specific case of the Pachinko Allocation document model, namely *Four-Level Pachinko Allocation* which captures correlations between words, and correlations between topic indices.

Formally, the Four-Level Pachinko Allocation model is a probabilistic generative model for collections of discrete data. In the context of document classification, PA assumes that each document can be represented as a collection of mixtures of latent topics, where each mixture is weighted by a mixture of latent super-topics. Solving for latent variables is achieved in [5] via Gibbs sampling which has the limitation that not only that the algorithm requires many passes of the data, but it also requires many passes of the individual word tokens. This algorithm is extremely memory intensive and does not scale well in greater problem settings such as large corpora. To tackle this, we introduce Batch Variational Bayes for PA and a novel extension to yield Online Variational Bayes for PA.

In this chapter, we outline in detail the Four-Level Pachinko Allocation Model, present Batch Variational Bayes for PA, introduce a novel Online Variational Bayes algorithm for PA, and finally, perform various sets of experiments and investigate the performance of mathematical document classification via Pachinko Allocation.

6.1 Four-Level Pachinko Allocation

In this section, we outline the Four-Level Pachinko Allocation model (from here we drop the "Four-Level") as described in [5]. Pachinko Allocation (PA) is a probabilistic generative model with a similar Dirichlet/Categorical distribution framework to LDA. In the context of document modelling, PA assumes that a document can be represented by a *collection* of mixtures of latent topics $\boldsymbol{\theta}_{d1}, \ldots, \boldsymbol{\theta}_{dS}$, where each topic mixture $\boldsymbol{\theta}_{di}$ is weighted according to a mixture of latent super-topics $\boldsymbol{\theta}_d^r$, where each topic is characterised by a distribution of words.

The generative process of PA has an extra hierarchical level of generation when compared to LDA. PA assumes that a document can be generated by first sampling a latent super-topic mixture and a collection of topic mixtures, then sampling super-topic and topic indices for each word according to these mixtures, then finally sampling a selection words according to these topics.

Formally, this model assumes that the number of latent super-topics S, the number of latent topics K, and the size of the vocabulary V are known and fixed. Furthermore, the document lengths N_d are Poisson distributed with parameter ξ. Finally, the K topics are Dirichlet distributed with smoothing parameter $\boldsymbol{\eta}$.

Similar to LDA, the Poisson assumption is not critical to anything that follows in this Chapter.

Given a Dirichlet priors $\boldsymbol{\alpha}^r$ on the super-topic mixtures, and $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_S$ on the topic mixtures. We outline the generative process of a document $\mathbf{w}_d$ under PA in Algorithm 9 where we highlight the new steps when compared to the generative process under LDA.

Algorithm 9 Generative process of a document under PA

Sample super-topic mixture $\boldsymbol{\theta}^r \sim \text{Dir}(\boldsymbol{\alpha}^r)$ ▷ New step
for each of the S super-topics **do** ▷ New step
  Sample topic mixture $\boldsymbol{\theta}_{di} \sim \text{Dir}(\boldsymbol{\alpha}_i)$
end for
Sample number of words $N_d \sim \text{Poisson}(\xi)$
for each of the N_d words **do**
  Sample word super-topic index $i = z_{dn} \sim \text{Cat}(\boldsymbol{\theta}^r_d)$ ▷ New step
  Sample word topic index $j = z'_{dn} \sim \text{Cat}(\boldsymbol{\theta}_{di})$
  Sample word index $\mathbf{w}_{dn} \sim \text{Cat}(\boldsymbol{\beta}_j)$
end for

Figure 6-1 shows an example of the super-topic and topic mixtures for three documents and three word and symbol topics under the PA model.

Under the generative process given by Algorithm 9, the joint distribution of a document $\mathbf{w}_d$ with super-topic mixture $\boldsymbol{\theta}^r_d$, topic mixtures $\boldsymbol{\theta}_{d1}, \ldots, \boldsymbol{\theta}_{dS}$, super-topic and topic indices $\mathbf{z}_d$ and $\mathbf{z}'_d$ respectively, and topics $\boldsymbol{\beta}$ is given by

$$\begin{aligned} p(\mathcal{H}, \mathbf{w}_d|\boldsymbol{\Theta}) = p(\boldsymbol{\theta}^r_d|\boldsymbol{\alpha}^r) \prod_{i=1}^{S} p(\boldsymbol{\theta}_{di}|\boldsymbol{\alpha}_i) \prod_{j=1}^{K} p(\boldsymbol{\beta}_j|\boldsymbol{\eta}) \\ \times \prod_{n=1}^{N_d} p(\mathbf{z}_{dn}|\boldsymbol{\theta}^r_d) p(\mathbf{z}'_{dn}|\mathbf{z}_{dn}, \boldsymbol{\theta}_d) p(\mathbf{w}_{dn}|\mathbf{z}'_{dn}, \boldsymbol{\beta}) \end{aligned} \tag{6.1}$$

where $\mathcal{H}$ denotes the set of latent variables, and $\boldsymbol{\Theta}$ denotes the set of model parameters. The factors $p(\boldsymbol{\theta}^r_d|\boldsymbol{\alpha}^r)$, $p(\boldsymbol{\theta}_{di}|\boldsymbol{\alpha}_i)$ and $p(\boldsymbol{\beta}_j|\boldsymbol{\eta})$ are given by the probability density function of the Dirichlet distribution, and the remaining factors $p(\mathbf{z}'_{dn}|z_{dn} = i, \boldsymbol{\theta}_d)$ and $p(\mathbf{w}_{dn}|z'_{dn} = j, \boldsymbol{\beta})$ are given by the probability density function of the Categorical distribution via $p(\mathbf{z}'_{dn}|\boldsymbol{\theta}_{di})$ and $p(\mathbf{w}_{dn}|\boldsymbol{\beta}_j)$ respectively.

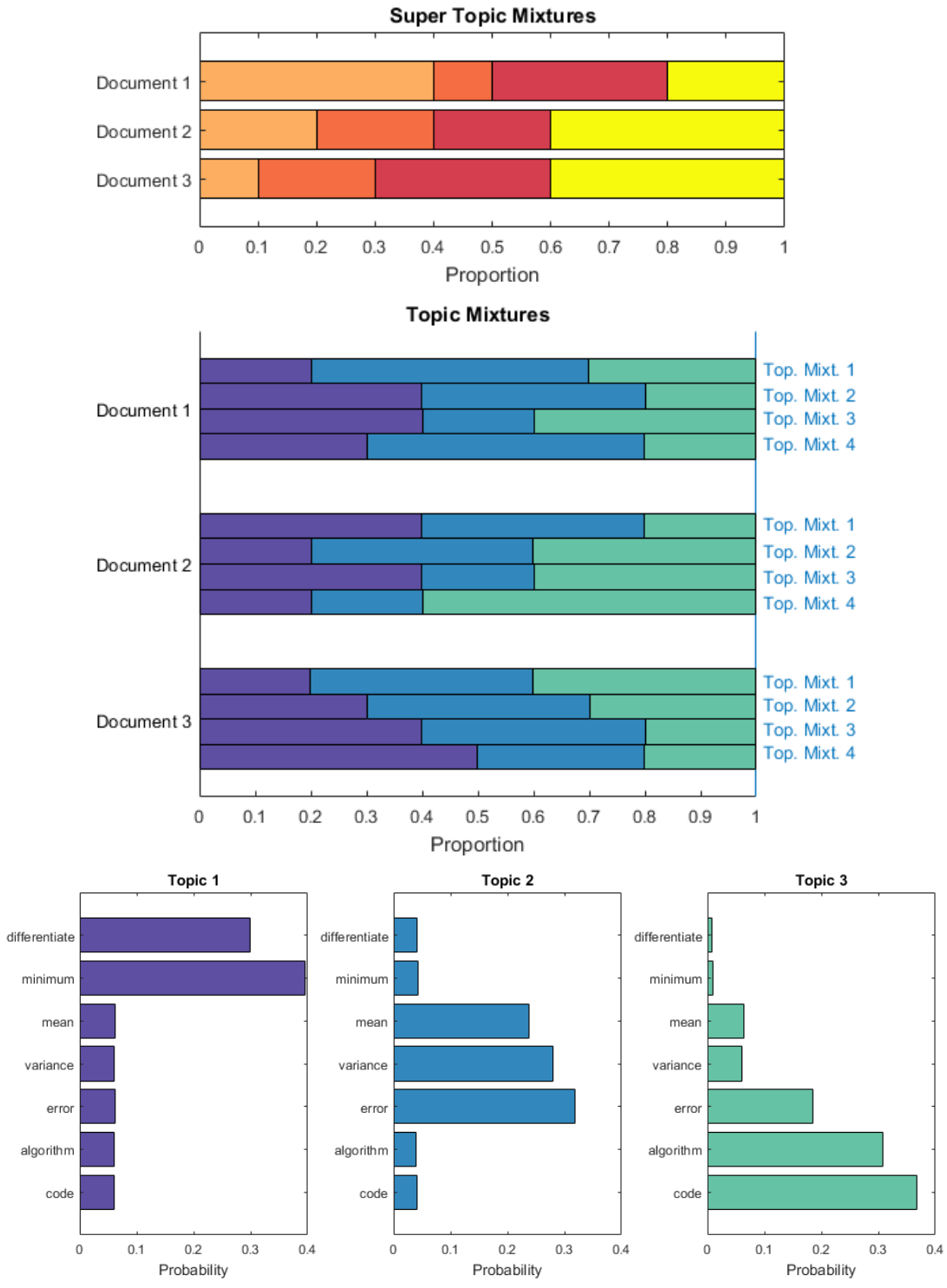

Figure 6-1: Example super-topic and topic mixtures with corresponding topics under PA.

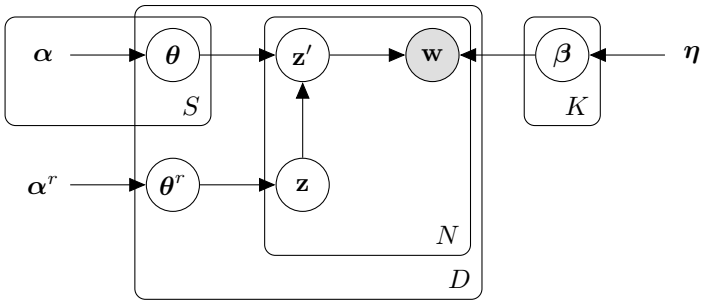

Figure 6-2: Graphical model representation of PA

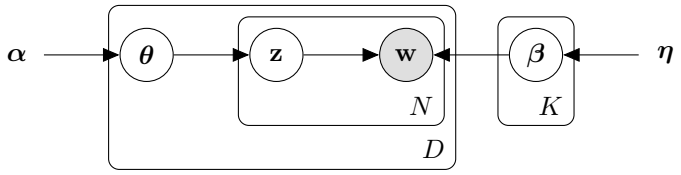

Figure 6-3: Graphical model representation of LDA

Figure 6-2 describes the PA model as a probabilistic graphical model and makes clear the four levels of the generative process:

1. The words are attributed by the word-level topic indices.
2. The topic indices are attributed by the word-level super-topic indices.
3. The super-topic indices are sampled according to the document-level super-topic mixture.
4. The super-topic mixture is sampled according to the corpus level Dirichlet prior.

In comparison, Figure 6-3 makes clear the extra level of PA, in particular, if we consider the case when $S = 1$, the super-topic specific variables become trivial, and we obtain the original LDA graphical model.

6.2 Inference

We now outline inference for Pachinko Allocation. In this section, we present we introduce Variational Bayes for PA using a similar approach to Variational Bayes

for LDA. Due to the similarities of the structure of PA and LDA, we discover that some of the components of the inference procedure are identical. We first introduce Batch Variational Bayes for PA, and later describe an adaptation and introduce a novel algorithm for Online Variational Bayes for PA.

We wish to solve for latent variables in the PA model; given the model parameters, we would like to determine the values of the latent variables which maximise the likelihood of a document $\mathbf{w}_d$ via the posterior distribution

$$p(\mathcal{H}|\mathbf{w}_d, \mathbf{\Theta}) = \frac{p(\mathcal{H}, \mathbf{w}_d|\mathbf{\Theta})}{p(\mathbf{w}_d|\mathbf{\Theta})}$$

As before, this posterior distribution is intractable to compute in general due to the coupling between the topic mixtures and the topics. In the next section, we outline variational inference methods to approximate this posterior.

6.2.1 Variational Inference

In [5], inference for PA is achieved via Gibbs Sampling; a powerful inference technique which iteratively samples the values of each latent variable while keeping remaining latent variables fixed until convergence. Gibbs Sampling is a very computationally demanding algorithm: Gibbs Sampling requires many repeated word level calculations and passes of the corpus, unlike Batch Variational Bayes, which requires repeated many document level calculations and passes of the corpus, and finally, Online Variational Bayes which still requires many repeated document level calculations but only a single pass of the corpus.

Gibbs Sampling has very high memory requirements since each of the word-level and document-level variables are read and written so frequently. In [43], the authors present a memory efficient sparse version of Gibbs Sampling for PA but still does not address the time requirements due to of the amount of word and document level updates. We tackle this problem by developing Online Variational Bayes for PA: a novel inference procedure for PA which requires only document level updates and a single pass of the corpus. Furthermore, by keeping the inference techniques similar (and where possible identical) for our document models in this thesis, we can focus more on testing the claims that mathematical document classification requires symbol data without influencing the results by

using different machine learning techniques.

Variational Bayes for generic topic models is outlined in [18], however, in this paper, the mathematics is dense and difficult to follow. In this section, we begin by deriving Variational Bayes for Pachinko Allocation from the ground up using the same strategy as deriving Variational Bayes for LDA. Recall that the strategy for variational inference is to obtain the tightest possible lower bound on the log-likelihood [33] by optimising this lower bound over a set of free variational parameters.

We begin by constructing a lower bound using the same strategy as in Variational Bayes for LDA. We consider a simpler version of the graphical model of PA in Figure 6-2 with the problematic nodes and edges removed and augment this model with a set of free variational parameters: we remove all edges, observed variables, and model parameters; we then augment the model with the free variational parameters $\boldsymbol{\gamma}^r$, $\boldsymbol{\gamma}$, $\boldsymbol{\phi}$, $\boldsymbol{\phi}'$, and $\boldsymbol{\lambda}$ with edges according to the variational distribution

$$q(\mathcal{H}|\mathcal{F}) = \prod_{j=1}^{K} q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j) \prod_{d=1}^{D} q_d(\mathcal{H}|\mathcal{F}) \tag{6.2}$$

where $\mathcal{F}$ denotes the set of free variational parameters, and q_d denotes the variational distribution of a single document given by

$$q_d(\mathcal{H}|\mathcal{F}) = q(\boldsymbol{\theta}_d^r|\boldsymbol{\gamma}_d^r) \prod_{i=1}^{S} q(\boldsymbol{\theta}_{di}|\boldsymbol{\gamma}_{di}) \prod_{n=1}^{N_d} q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn}) q(\mathbf{z}'_{dn}|\boldsymbol{\phi}'_{dn}) \tag{6.3}$$

The factors $q(\boldsymbol{\theta}^r|\boldsymbol{\gamma}^r)$, $q(\boldsymbol{\theta}_i|\boldsymbol{\gamma}_i)$, and $q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j)$ are given by the probability density function of the Dirichlet distribution, and $q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn})$ and $q(\mathbf{z}'_{dn}|\boldsymbol{\phi}'_{dn})$ are given by the probability density function of the Categorical distribution. Figure 6-4 describes the full variational distribution as a probabilistic graphical model.

We obtain a lower bound of the log-likelihood using Jensen's inequality via the same method as deriving the lower bound for the log-likelihood of a document under LDA. Furthermore, we arrive at the same lower bound as a function of expectations over the variational distribution q given by the inequality

$$\log p(\mathbf{w}|\boldsymbol{\Theta}) \geq \mathbb{E}_q[\log p(\mathcal{H}, \mathbf{w}|\boldsymbol{\Theta})] - \mathbb{E}_q[\log q(\mathcal{H}|\mathcal{F})] \tag{6.4}$$

We let $\mathcal{L}$ denote the right hand side of Equation (6.4) as a function of the free Vari-

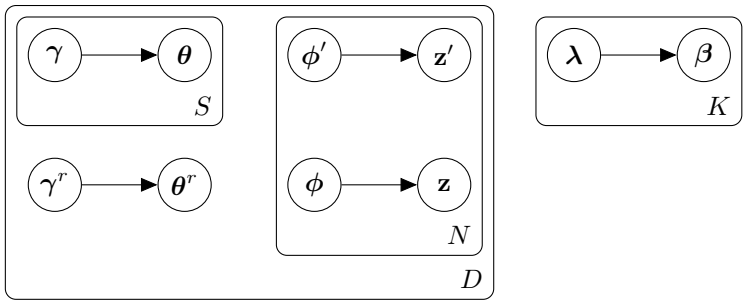

Figure 6-4: Graphical model representation of the variational distribution of PA

ational parameters $\mathcal{F}$ given the model parameters $\mathbf{\Theta}$ and call this the *Evidence Lower Bound*. We rewrite $\mathcal{L}$ as a sum of expectations using the factorisations of p and q and arrive at

$$\mathcal{L}(\mathcal{F};\mathbf{\Theta}) = \sum_{d=1}^{D} L_d(\mathcal{F};\mathbf{\Theta}) + \mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] - \mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] \tag{6.5}$$

where L_d, the contribution of the dth document to the Evidence Lower Bound is given by

$$\begin{aligned} L_d(\mathcal{F};\mathbf{\Theta}) = {} & \mathbb{E}_q[\log p(\boldsymbol{\theta}_d^r|\boldsymbol{\alpha}^r)] + \mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] + \mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d^r)] \\ & + \mathbb{E}_q[\log p(\mathbf{z}'_d|\mathbf{z}_d,\boldsymbol{\theta}_d)] + \mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}'_d,\boldsymbol{\beta}_d)] - \mathbb{E}_q[\log q(\boldsymbol{\theta}_d^r|\boldsymbol{\gamma}_d^r)] \\ & - \mathbb{E}_q[\log q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)] - \mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)] - \mathbb{E}_q[\log q(\mathbf{z}'_d|\boldsymbol{\phi}')] \end{aligned} \tag{6.6}$$

We now have a lower bound on the log-likelihood of the corpus given an arbitrary variational distribution q. We now wish to minimise the difference between the log-likelihood and this lower bound. It can be verified that this difference is indeed the KL divergence between the variational posterior probability $q(\mathcal{H}|\mathcal{F})$ and the true posterior probability $p(\mathcal{H}|\mathbf{w},\mathbf{\Theta})$. Finally, by rewriting the log-likelihood as a function of the free variational parameters via the Evidence Lower Bound and this KL divergence, we arrive at

$$\log p(\mathbf{w}|\mathbf{\Theta}) = \mathcal{L}(\mathcal{F};\mathbf{\Theta}) + \mathrm{D}(q(\mathcal{H}|\mathcal{F}) \,\|\, p(\mathcal{H}|\mathbf{w},\mathbf{\Theta}))$$

We see that minimising the KL divergence between the variational posterior

probability and the true posterior probability is equivalent to maximising $\mathcal{L}$ via the free variational parameters. We now outline the steps for maximising the Evidence Lower Bound.

The expectations in Equation (6.5) each take the form of the expectations outlined in Appendix B. Using the expanded forms of these expectations, we express the terms of Equation (6.5) as follows. Firstly, we discover that the expectations over the topics $\boldsymbol{\beta}$ are identical to the same expectations under LDA. Appendices B.1 and B.2 show that the expectations over the super-topic mixtures $\boldsymbol{\theta}_d^r$ expand to the summations

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d^r|\boldsymbol{\alpha}^r)] = \log\Gamma\left(\sum_i \alpha_i^r\right) + \sum_{i=1}^{S}((\alpha_i^r - 1)\mathbb{E}_q[\log\theta_{di}^r|\boldsymbol{\gamma}_d^r] - \log\Gamma(\alpha_i^r))$$
$$\mathbb{E}_q[\log q(\boldsymbol{\theta}_d^r|\boldsymbol{\gamma}_d^r)] = \log\Gamma\left(\sum_i \gamma_{di}^r\right) + \sum_{i=1}^{S}((\gamma_{di}^r - 1)\mathbb{E}_q[\log\theta_{di}^r|\boldsymbol{\gamma}_d^r] - \log\Gamma(\gamma_i^r))$$

where we emphasise the dependency on $\boldsymbol{\gamma}_d^r$ in the inner expectations. Appendices B.2 and B.3 show the expectations over the topic mixtures $\boldsymbol{\theta}_d$ expand to

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] = \sum_{i=1}^{S}\left(\log\Gamma\left(\sum_j \alpha_{ij}\right) + \sum_{j=1}^{K}((\alpha_{ij}-1)\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_{di}] - \log\Gamma(\alpha_{ij}))\right)$$
$$\mathbb{E}_q[\log q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)] = \sum_{i=1}^{S}\left(\log\Gamma\left(\sum_j \gamma_{dij}\right) + \sum_{j=1}^{K}((\gamma_{dij}-1)\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_{di}] - \log\Gamma(\gamma_{dij}))\right)$$

where we emphasise the dependency on $\boldsymbol{\gamma}_d$ in the inner expectations. Appendices B.4 and B.5 show the expectations over the super-topic indices $\mathbf{z}_d$ expand and rearrange to give the summations

$$\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d^r)] = \sum_{n=1}^{N_d}\sum_{i=1}^{S}\phi_{dni}\mathbb{E}_q[\log\theta_{di}^r|\boldsymbol{\gamma}_d^r]$$
$$\mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)] = \sum_{n=1}^{N_d}\sum_{i=1}^{S}\phi_{dni}\log\phi_{dni}$$

where we emphasise the dependency on $\boldsymbol{\gamma}_d^r$ in the inner expectations. Similarly,

Appendix B.5 also shows that

$$\mathbb{E}_q[\log q(\mathbf{z}'|\boldsymbol{\phi}'_d)] = \sum_{n=1}^{N_d}\sum_{j=1}^{K}\phi'_{dnj}\log\phi'_{dnj}$$

Finally, Appendix B.6 shows the remaining expectation over the topic indices $\mathbf{z}'_d$ expands and rearranges to the summation

$$\mathbb{E}_q[\log p(\mathbf{z}'_d|\mathbf{z}_d,\boldsymbol{\theta}_d)] = \sum_{n=1}^{N_d}\sum_{i=1}^{S}\phi_{dni}\sum_{j=1}^{K}\phi'_{dnj}\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_d]$$

where we emphasise the dependency on $\boldsymbol{\gamma}_d$ in the inner expectations.

By plugging in these expectations into Equation (6.5), we now have the lower bound $\mathcal{L}$ as a function of the model parameters and the free variational parameters. We now outline the process of maximising this lower bound via the variational parameters.

6.2.1.1 Document Level Updates

In this section, we describe the methods of maximising $\mathcal{L}$ with respect to each of the document level variational parameters $\boldsymbol{\phi}_d$, $\boldsymbol{\phi}'_d$, $\boldsymbol{\gamma}^r_d$ and $\boldsymbol{\gamma}_d$.

Variational Categorical Parameters We first maximise $\mathcal{L}$ with respect to the Variational Categorical parameters $\boldsymbol{\phi}_{dn}$. We maximise each entry of $\boldsymbol{\phi}_{dn}$ individually and use Lagrange multipliers to enforce the constraint that the entries in $\boldsymbol{\phi}_{dn}$ must sum to one. Retaining only the terms of $\mathcal{L}$ containing ϕ_{dni} and adding the appropriate Lagrange multiplier Λ_{dn} yields

$$\mathcal{L}_{[\phi_{dni}]} = \phi_{dni}\left(\mathbb{E}_q[\log\theta^r_{di}|\boldsymbol{\gamma}^r_d] + \sum_{j=1}^{K}\phi'_{dnj}\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_{di}] - \log\phi_{dni}\right) + \Lambda_{dn}\left(\sum_{i'}\phi_{dni'} - 1\right)$$

taking partial derivatives with respect to ϕ_{dni} yields

$$\frac{\partial\mathcal{L}_{[\phi_{dni}]}}{\partial\phi_{dni}} = \mathbb{E}_q[\log\theta^r_{di}|\boldsymbol{\gamma}^r_d] + \sum_{j=1}^{K}\phi'_{dnj}\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_{di}] - \log\phi_{dni} - 1 + \Lambda_{dn}$$

Using Equation (4.5) to expand the expected logs and setting this derivative to zero yields the maximising value of ϕ_{dni} at

$$\phi_{dni} \propto \exp\left\{\mathbb{E}_q[\log \theta^r_{di}|\boldsymbol{\gamma}^r_d] + \sum_{j=1}^{K} \phi'_{dnj}\mathbb{E}_q[\log \theta_{dij}|\boldsymbol{\gamma}_{di}]\right\}$$

where $\boldsymbol{\phi}_{dn}$ is normalised to sum to one.

We now maximise $\mathcal{L}$ with respect to the Variational Categorical parameters $\boldsymbol{\phi}'_{dn}$ where we again perform element-wise maximisation and use Lagrange multipliers to enforce the constraint that entries of $\boldsymbol{\phi}'_{dn}$ must sum to one. Retaining only the terms of $\mathcal{L}$ containing ϕ'_{dnj} and adding the appropriate Lagrange multiplier Λ_{dn} yields

$$\mathcal{L}_{[\phi'_{dnj}]} = \phi'_{dnj}\left(\sum_i \phi_{dni}\mathbb{E}_q[\log \theta_{dij}|\boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j] - \log \phi'_{dnj}\right) + \Lambda_{dn}\left(\sum_{j'} \phi'_{dnj'} - 1\right)$$

taking partial derivatives with respect to ϕ'_{dnj} yields

$$\frac{\partial \mathcal{L}_{[\phi'_{dnj}]}}{\partial \phi'_{dnj}} = \sum_{i=1}^{S} \phi_{dni}\mathbb{E}_q[\log \theta_{dij}|\boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j] - \log \phi'_{dnj} - 1 + \Lambda_n$$

Finally, setting this derivative to zero and solving yields the maximising value of ϕ'_{dnj} at

$$\phi'_{dnj} \propto \exp\left\{\sum_{i=1}^{S} \phi_{dni}\mathbb{E}_q[\log \theta_{dij}|\boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\right\}$$

where $\boldsymbol{\phi}'_{dn}$ is normalised to sum to one.

Variational Dirichlet Parameters We now maximise $\mathcal{L}$ with respect to the Variational Dirichlet parameter γ^r_d. We maximise each entry of $\boldsymbol{\gamma}^r_d$ individually with no constraints to enforce. By retaining only the terms of $\mathcal{L}$ containing γ^r_{di} and simplifying we obtain

$$\mathcal{L}_{[\gamma^r_{di}]} = \sum_{i'=1}^{S} \mathbb{E}_q[\log \theta^r_{di'}|\boldsymbol{\gamma}^r_d]\left(\alpha^r_{i'} - \gamma^r_{di} + \sum_{n=1}^{N_d} \phi_{dni'}\right) - \log\Gamma\left(\sum_{i'} \gamma^r_{di'}\right) + \log\Gamma(\gamma^r_{di})$$

Taking partial derivatives with respect to γ_{di}^r yields

$$\frac{\partial \mathcal{L}_{[\gamma_{di}^r]}}{\partial \gamma_{di}^r} = \Psi'(\gamma_{di}^r)\left(\alpha_i^r - \gamma_{di}^r + \sum_{n=1}^{N_d}\phi_{dni}\right) - \Psi'\left(\sum_{i'}\gamma_{di'}^r\right)\sum_{i'=1}^{S}\left(\alpha_{i'}^r - \gamma_{di'}^r + \sum_{n=1}^{N_d}\phi_{dni'}\right)$$

Setting this derivative to zero yields the maximising value of γ_{di}^r at

$$\gamma_{di}^r = \alpha_i^r + \sum_{n=1}^{N_d}\phi_{dni}$$

We now maximise $\mathcal{L}$ with respect to $\boldsymbol{\gamma}_{di}$, the where again we maximise each entry of $\boldsymbol{\gamma}_{di}$ individually with no constraints to enforce. Retaining only the terms of $\mathcal{L}$ containing γ_{dij} and simplifying we have

$$\begin{aligned}\mathcal{L}_{[\gamma_{dij}]} = \sum_{j'=1}^{K}\mathbb{E}_q[\log\theta_{dij'}|\boldsymbol{\gamma}_{di}]\left(\alpha_{ij'} - \gamma_{dij'} + \sum_{n=1}^{N_d}\phi_{dni}\phi'_{dnj'}\right) \\ - \log\Gamma\left(\sum_{j'}\gamma_{dij'}\right) + \log\Gamma(\gamma_{dij})\end{aligned}$$

Taking partial derivatives with respect to γ_{dij} yields

$$\begin{aligned}\frac{\partial \mathcal{L}_{[\gamma_{dij}]}}{\partial \gamma_{dij}} = \Psi'(\gamma_{dij})\left(\alpha_{ij} - \gamma_{dij} + \sum_{n=1}^{N_d}\phi_{dni}\phi'_{dnj}\right) \\ - \Psi'\left(\sum_{j'}\gamma_{dij'}\right)\sum_{j'=1}^{K}\left(\alpha_{ij'} - \gamma_{dij'} + \sum_{n=1}^{N_d}\phi_{dni}\phi'_{dnj'}\right)\end{aligned}$$

Finally, setting this derivative to zero yields the maximising value of γ_{dij} at

$$\gamma_{dij} = \alpha_{ij} + \sum_{n=1}^{N_d}\phi_{dni}\phi'_{dnj}$$

We now have update rules for the variational parameters which we require for the document level variational inference procedure. Since the update rules for the document level variational parameters are dependent on one another, full variational inference requires iterating through these update rules until convergence. We summarise the document level variational updates in Algorithm 10.

Algorithm 10 Document level variational inference for Pachinko Allocation

Initialise γ^r_d, γ_d randomly

repeat

Set $\boldsymbol{\phi}_{dni} \propto \exp\left\{\mathbb{E}_q[\log \theta^r_{di}|\boldsymbol{\gamma}^r_d] + \sum_j \phi'_{dnj}\mathbb{E}_q[\log \theta_{dij}|\boldsymbol{\gamma}_{di}]\right\}$

Set $\boldsymbol{\phi}'_{dnj} \propto \exp\left\{\sum_i \phi_{dni}\mathbb{E}_q[\log \theta_{dij}|\boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\right\}$

Set $\boldsymbol{\gamma}^r_{di} = \alpha^r_i + \sum_n \phi_{dni}$

Set $\boldsymbol{\gamma}_{dij} = \alpha_{ij} + \sum_n \phi_{dni}\phi'_{dnj}$

until Convergence of $\boldsymbol{\gamma}^r_d$, $\boldsymbol{\gamma}_d$

Similar to LDA, we may optimise for a single fixed document $\mathbf{w}_d$. Therefore, we can use the procedure that yields the optimised values of $\boldsymbol{\phi}'_d$, to approximate the topic mixture representation of a given document. In particular, we evaluate the expected value of the normalised frequency count of the topic indices $\mathbf{z}'_{dn}$ under the variational distribution q:

$$\bar{\boldsymbol{\phi}}'_d = \mathbb{E}_q\left[\frac{1}{N_d}\sum_{n=1}^{N_d}\mathbf{z}'_{dn}\right] = \frac{1}{N_d}\sum_{n=1}^{N_d}\boldsymbol{\phi}'_{dn}$$

6.2.1.2 Corpus Level Updates

We now have a variational inference procedure for approximating the document level variational parameters. We now derive the variational methods for approximating the variational parameters $\boldsymbol{\lambda}$ and the Dirichlet priors $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}$ which maximise the marginal log-likelihood of the data.

We have already shown that there is a tractable lower bound on the log-likelihood, and we can further maximise this lower bound via the model parameters $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}$ and the variational parameters $\boldsymbol{\lambda}$. Therefore, we can derive a full variational EM procedure that yields the optimised values for $\boldsymbol{\alpha}^r$, $\boldsymbol{\alpha}$, and $\boldsymbol{\lambda}$.

We optimise $\boldsymbol{\lambda}$ by using the same method for optimising $\boldsymbol{\lambda}$ under LDA. We discover a near identical update rule, the only difference being notational: we exchange $\boldsymbol{\phi}$ with $\boldsymbol{\phi}'$ since $\boldsymbol{\phi}$ now represents the Variational Categorical parameter on the super-topics indices under PA and the $\boldsymbol{\lambda}$ updates require the Variational Categorical parameter on the topic indices. That is, we have a maximising value

of λ_{jv} at

$$\lambda_{jv} = \eta_v + \sum_{d=1}^{D} \sum_{n=1}^{N_d} w_{dnv} \phi'_{dnj}$$

We now maximise the lower bound $\mathcal{L}$ via the Dirichlet parameters $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}$. We achieve this via the same method as approximating $\boldsymbol{\alpha}$ under LDA. In particular, we optimise $\boldsymbol{\alpha}^r$ using the values of the $\boldsymbol{\gamma}_d^r$'s, and optimise each $\boldsymbol{\alpha}_i$ using the values of the $\boldsymbol{\gamma}_{di}$'s.

Firstly, we optimise $\boldsymbol{\alpha}^r$ using [36] via the update rule

$$\boldsymbol{\alpha}^r \leftarrow \boldsymbol{\alpha}^r - \tilde{\boldsymbol{\alpha}}^r(\boldsymbol{\gamma}^r)$$

where $\tilde{\boldsymbol{\alpha}}^r$ is the inverse of the Hessian H multiplied by the gradient $\mathbf{g}$ as a function of $\boldsymbol{\gamma}^r$, where

$$g_i = D\left(\Psi\left(\sum_{i'} \alpha_i^r\right) - \Psi(\alpha_i^r)\right) + \sum_{d=1}^{D} \mathbb{E}_q[\log \theta_{di}^r | \boldsymbol{\gamma}_d^r]$$

$$H_{ii'} = \delta(i, i') D\Psi'(\alpha_i^r) - \Psi'\left(\sum_{i''} \alpha_{i''}^r\right)$$

Finally, we optimise each $\boldsymbol{\alpha}_i$ using [36] with the update rule

$$\boldsymbol{\alpha}_i \leftarrow \boldsymbol{\alpha}_i - \tilde{\boldsymbol{\alpha}}_i(\boldsymbol{\gamma})$$

where $\tilde{\boldsymbol{\alpha}}_i$ is the inverse of the Hessian H_i multiplied by the gradient $\mathbf{g}_i$ as a function of the $\boldsymbol{\gamma}_{di}$'s, where

$$g_{ij} = D\left(\Psi\left(\sum_{j'} \alpha_{ij}\right) - \Psi(\alpha_{ij})\right) + \sum_{d=1}^{D} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}]$$

$$H_{ijj'} = \delta(j, j') D\Psi'(\alpha_{ij}) - \Psi'\left(\sum_{j''} \alpha_{ij''}\right)$$

We now have the required document level and corpus level updates necessary for the full Batch Variational Bayes algorithm for Pachinko Allocation. The outline of the variational EM procedure is as follows:

- E-step: For each document, find the optimised values of the variational pa-

rameters $\boldsymbol{\gamma}_d^r$, $\boldsymbol{\gamma}_d$, $\boldsymbol{\phi}_d$, and $\boldsymbol{\phi}'_d$ using the document level variational algorithm.

- M-step: Maximise the resulting lower bound on the log-likelihood via the parameters $\boldsymbol{\alpha}^r$, $\boldsymbol{\alpha}$, and $\boldsymbol{\lambda}$ using the corpus level updates.

We summarise the full Variational Bayes inference procedure for Pachinko Allocation on a corpus of documents in Algorithm 11. We apply the document level variational procedure for each document in the corpus, and update the corpus level parameters after each pass of the data.

Algorithm 11 Batch Variational Bayes for Pachinko Allocation

Initialise $\boldsymbol{\lambda}$ randomly
repeat
  for $d = 1, \ldots, D$ **do**
    Initialise γ_d^r, γ_d randomly
    repeat
      Set $\boldsymbol{\phi}_{dni} \propto \exp\Big\{\mathbb{E}_q[\log \theta_{di}^r | \boldsymbol{\gamma}_d^r] + \sum_j \phi'_{dnj} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}]\Big\}$
      Set $\boldsymbol{\phi}'_{dnj} \propto \exp\{\sum_i \phi_{dni} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}} | \boldsymbol{\lambda}_j]\}$
      Set $\boldsymbol{\gamma}_{di}^r = \alpha_i^r + \sum_n \phi_{dni}$
      Set $\boldsymbol{\gamma}_{dij} = \alpha_{ij} + \sum_n \phi_{dni} \phi'_{dnj}$
    until Convergence of $\boldsymbol{\gamma}_d^r$, $\boldsymbol{\gamma}_d$
  end for
  Set $\lambda_{jv} = \eta_v + \sum_d \sum_n w_{dnv} \phi'_{dnj}$
until Convergence of $\mathcal{L}$

6.2.2 Online Variational Inference

Algorithm 11 is *batch* Variational Bayes algorithm; it requires a full pass of the corpus at each iteration. We now present Online Variational Bayes for Pachinko Allocation which requires only a single pass of the corpus. In particular, based on [6], we introduce a novel adaptation of the Batch Variational Bayes algorithm for PA so that the variational parameter $\boldsymbol{\lambda}$ is updated at each iteration. We update $\boldsymbol{\lambda}$ using a weighted average of its previous value and the optimal value according to the current value of $\boldsymbol{\phi}'$ via the weighting parameter $\rho_d = (\tau_0 + d)^{-\kappa}$ where τ_0 denotes the offset and κ denotes the decay as described in Chapter 4.

We describe the Online Variational Bayes algorithm for Pachinko Allocation in Algorithm 12.

Algorithm 12 Online Variational Bayes for Pachinko Allocation

Define $\rho_d := (\tau_0 + d)^{-\kappa}$
Initialise $\boldsymbol{\lambda}$ randomly
for $d = 1, 2, \ldots$ **do**
  Initialise γ_d^r, γ_d randomly
  repeat
    Set $\boldsymbol{\phi}_{dni} \propto \exp\left\{\mathbb{E}_q[\log \theta_{di}^r | \boldsymbol{\gamma}_d^r] + \sum_j \phi'_{dnj} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}]\right\}$
    Set $\boldsymbol{\phi}'_{dnj} \propto \exp\left\{\sum_i \phi_{dni} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}] + \mathbb{E}_q\left[\log \beta_{j w_{dn}} | \boldsymbol{\lambda}_j\right]\right\}$
    Set $\boldsymbol{\gamma}_{di}^r = \alpha_i^r + \sum_n \phi_{dni}$
    Set $\boldsymbol{\gamma}_{dij} = \alpha_{ij} + \sum_n \phi_{dni} \phi'_{dnj}$
  until Convergence of $\boldsymbol{\gamma}_d^r$, $\boldsymbol{\gamma}_d$
  Set $\tilde{\lambda}_{jv} = \eta_v + D \sum_n w_{dnv} \phi'_{dnj}$
  Set $\boldsymbol{\lambda} = (1 - \rho_d)\boldsymbol{\lambda} - \rho_d \tilde{\boldsymbol{\lambda}}$
end for

6.3 Document Classification

In this section, we outline document classification via Pachinko Allocation. By design, the framework is almost identical to classification via Latent Dirichlet Allocation as described in Section 4.4. In particular, we describe a semi-supervised classifier trained on a partially labelled collection of documents via their topic mixture representations.

6.3.1 Framework

We outline the complete framework of document classification via Pachinko Allocation. As before, we use the machine learning framework described in 3: we break down the process into the unsupervised layer which uses PA to model the documents as topic mixtures, and the supervised layer which trains a supervised multi-label classifier on a collection of these labelled topic mixtures.

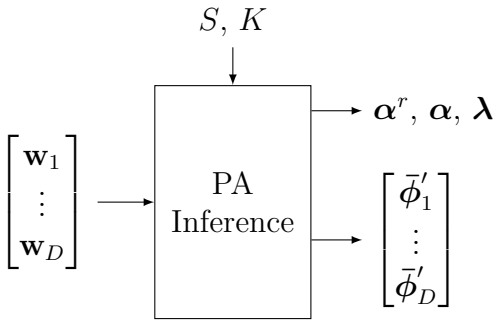

Figure 6-5: The Unsupervised Layer using PA

Document Modelling

Given a partially labelled corpus and choices of S and K, we use the PA parameter estimation methods to approximate the Dirichlet priors $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}$, the variational parameter $\boldsymbol{\lambda}$ on the topics, and the topic mixture representations $\bar{\phi}'_d$ of the documents. Figure 6-5 shows the document modelling step (the unsupervised layer) of the machine learning framework using PA.

Supervised Training

The supervised layer is identical to the ones described in LDA and DLDA. Using the labelled topic mixture representations of the documents obtained from the unsupervised layer as feature vectors, we train a supervised classifier. Again, we may use any discriminative supervised classification methods for this step. We continue to use nearest neighbour methods which we describe in Appendix C.2.

Classification

We now have all the tools necessary for classification via PA. The unsupervised layer provides the required parameters to obtain the topic mixture representation $\bar{\phi}'$ of an unseen document. The supervised layer provides the document classifier f which will output the predicted set of labels $\mathbf{c}$ for this document. Figure 6-6 outlines the classification process of a previously unseen document.

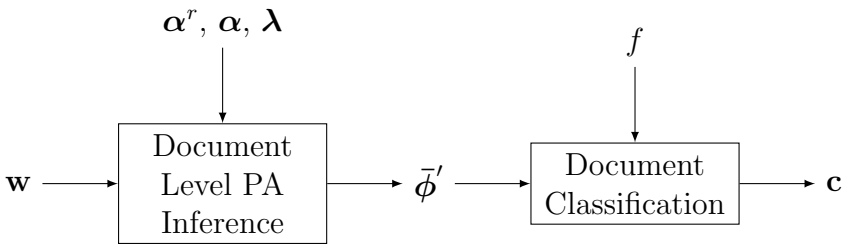

Figure 6-6: Classification process via PA

Remark

In the above framework, we do not incorporate the values of $\boldsymbol{\theta}^r$ in the classification step. For uncertain classification instances, for example, when the binary classifiers scores are close to the decision threshold, the inclusion of the super-topic mixtures may present useful discriminative characteristics. In the framework above, we classify documents according to the proportions of the word topics only. Given that the approximated values of the topics $\boldsymbol{\beta}$ depend on the values of $\boldsymbol{\theta}^r$ through $\boldsymbol{\theta}$, the features may already encapsulate sufficient information on the super-topics and this may have little effect on classification. The effect of including the values of $\boldsymbol{\theta}^r$ in the classification process is the subject for future work.

To summarise, PA provides another fast filtering algorithm for feature selection for document classification. In particular, PA allows us to transform a document of possibly many words to a mixture of a much smaller number of latent topics. Since PA takes into account the super-topic structures of the documents, the approximated topics are potentially less prone to overlap; PA controls correlations between topics at the super-topic level. Therefore, PA provides a means of representing documents as a mixture of latent topics with stronger discriminative properties compared to the LDA topic mixture representation.

6.4 Experimental Results

We now evaluate the performance of mathematical document classification via PA. In particular, we observe what effect the choice of both the number of super-topics S and the number of topics K has on classification performance, and furthermore, we determine the optimised Labelling F-Score by adjusting the decision boundaries on the supervised classifiers. Finally, we investigate the best performing classifiers in detail and study the confusion between subject areas.

6.4.1 Preliminary Experiments

To get an idea of how classification via PA behaves, we run our experiments for a selection of values of S and K and observe the effect it has on classification performance. For our preliminary experiments, we use a similar set-up as LDA in Section 4.5. We perform Pachinko Allocation on our training data using the values of $S \in \{1, 5, 10, 25, 50, 100, 200\}$ and $K \in \{10, 25, 50, 100, 200\}$. For the supervised layer, we train an ensemble of Nearest Neighbour Classifiers using five nearest neighbours, inverse distance weighting and the χ^2 distance metric. Note that when $S = 1$, we have an LDA model and achieve same classification performance. We ignore this result when selecting the best parameters.

Figure 6-7 shows the effect of the choice of values $S \in \{1, 5, 10, 25, 50, 100, 200\}$ and $K \in \{10, 25, 50, 100, 200\}$ on the Labelling F-Score of the Online PA classifier on the multi-labelled and uni-labelled datasets. Each experiment is repeated sixteen times on different random test and train partitions of the data.

Effect of S

We observe that introducing more super-topics has an adverse impact on performance in general. We observe a fairly consistent decrease in performance as the value of S increases. In particular, we see the highest classification performance when $S = 10$. For low values of K, the choice of S has little impact on the classification performance.

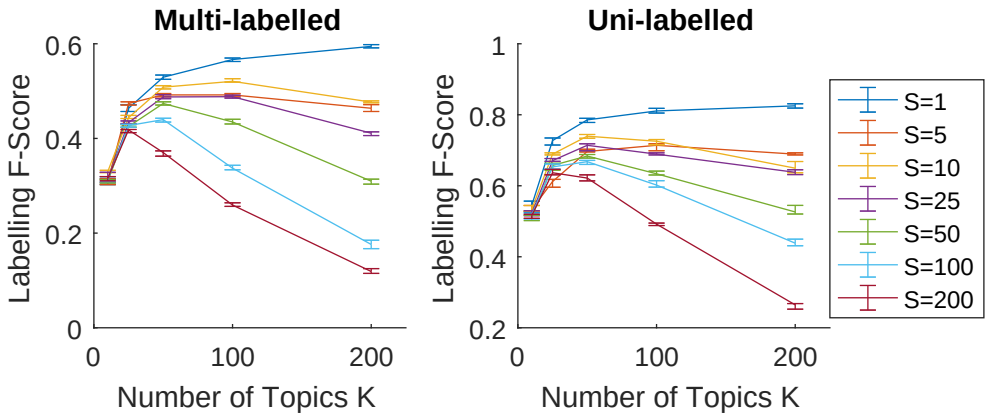

Figure 6-7: Classification performance via PA

Effect of K

Given low values of S, we see an increase in performance as K increases, up to the $K = 50$, $K = 100$ mark with slight declines in performance beyond. Unlike LDA and DLDA, we do not see the performance levelling off; we see obvious peaks with steady declines.

To summarise, on both datasets we achieve the highest Labelling F-Score with $S = 10$. Furthermore, we achieve the highest Labelling F-Score with $K = 100$ and $K = 50$ on the multi-labelled and uni-labelled datasets respectively. We now look in detail at the outputs of one of the classifiers in each of these cases.

Precision/Recall Trade-off

We now look at the micro-averaged precision and recall of the classification results. Over the sixteen experiments using the optimal parameter settings, we observe a median micro-averaged precision and recall of 84.13% and 73.97% respectively on the uni-labelled dataset and 76.33% and 40.00% respectively on the multi-labelled dataset. Similar to before, this suggests the binary classifiers are again too harsh when it comes to yielding positive classifications. We rerun our experiences on these classifiers with different settings for fixed decision thresholds and discover that we achieve optimal performance using the decision threshold of 0.3. Table 6.1 outlines the optimal parameter settings and performance of the classifiers on each dataset.

	Multi-labelled	Uni-labelled
S	10	10
K	50	100
Threshold	0.3	0.3
Labelling F-Score		
(median)	53.28%	79.07%
(maximum)	59.58%	80.00%
Micro-averaged F-Score		
(median)	53.29%	79.17%
(maximum)	57.69%	80.40%

Table 6.1: Optimised Classification Performance via Online Pachinko Allocation

6.4.2 Confusion

In this section, we investigate the per-class confusion of our classifier and furthermore, briefly compare performance to classification via LDA. Later, in Chapter 8, we will collect the results and compare all models presented in this thesis.

Firstly, Figure 6-8 shows a confusion matrix highlighting the per-class true positive rates on the diagonal, per-class false negative rates broken down by false positives over the other classes on the off-diagonal, and finally, per-class null classification rates on the final column. Compared with LDA, we notice a small decline in performance in the top performing categories, but a strong decline in performance in the poor performing categories. Furthermore, we see strong confusion between subject areas. In particular, we see 43.3% of documents labelled as "Ordinary differential equations" are misclassified as "Partial differential equations".

We now construct the equivalent confusion matrix for multi-labelled classification. Again, due to the size of the confusion matrix, we only present three sections of the matrix highlighting the areas of strongest and weakest performance. We present the full heat-map of confusion in Appendix D.

Figure 6-9 shows sections of the confusion matrix highlighting the strongest per-class true positive rates; the upper-left section. Here we see similar true positive and false negative rates to classification via LDA. Again, we see a slight increase in true positive rates amongst the top performing categories, but a small decrease elsewhere. Furthermore, we again see a general wash of misclassification

Figure 6-8: Per-class confusion of classification via PA on the uni-labelled dataset

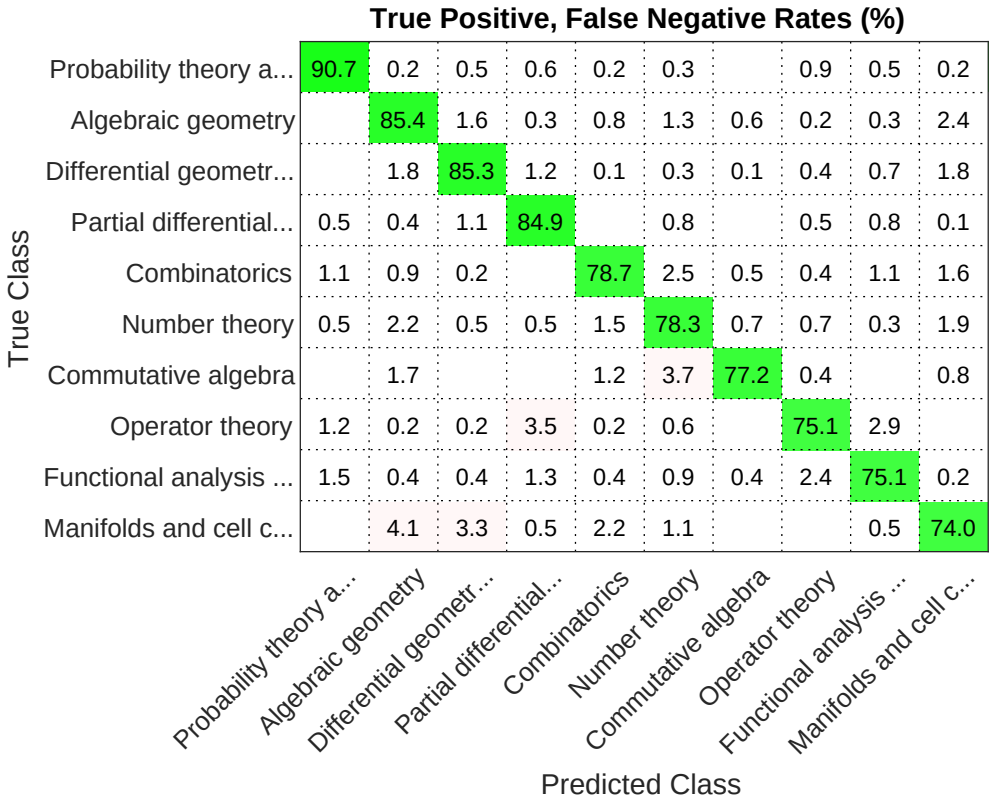

Figure 6-9: Per-class confusion of classification via PA on the multi-labelled dataset - Upper-left section.

amongst these top ten categories, but this time with slightly higher rates.

Figure 6-10 shows sections of the confusion matrix highlighting the strongest confusion between subject areas; the lower-left section. We again see that there are high levels of confusion between related subject areas such as "Integral equations" and "Partial differential equations". However, we are also seeing stronger confusion between non-related models. In particular, we see that the subject areas "Geometry" and "K-theory" are being false classified as many different areas.

Finally, Figure 6-11 shows sections of the confusion matrix highlighting the weakest per-class true positive rates; the lower-right section. Here we see a similar range true positive rates compared to classification via LDA, and furthermore, we see significantly lower rates of null classifications. Here we see that classification via PA administers a trade-off between false positive rates and null classification

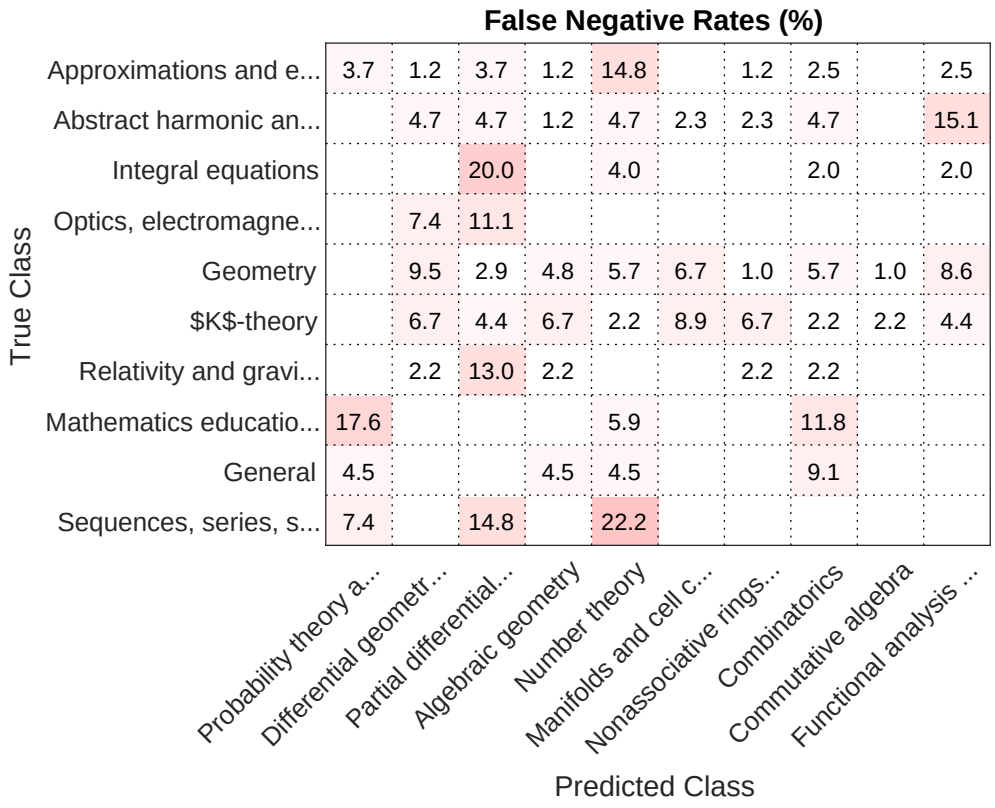

Figure 6-10: Per-class confusion of classification via PA on the multi-labelled dataset - Lower-left section.

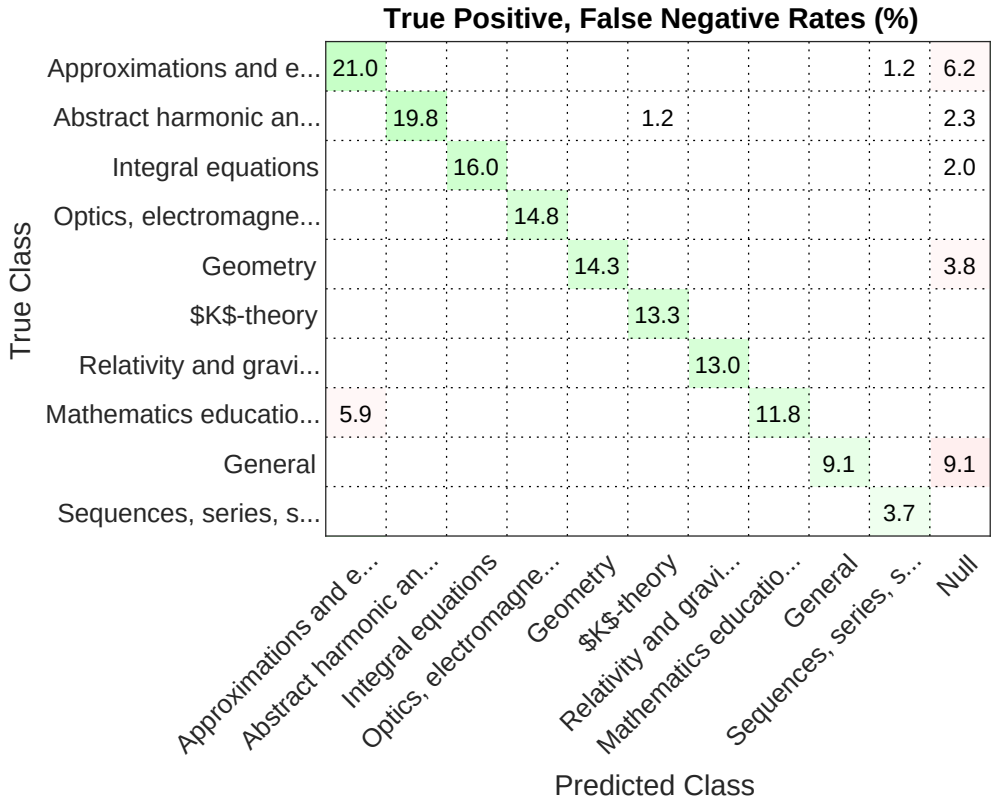

Figure 6-11: Per-class confusion of classification via PA on the multi-labelled dataset - Lower-right section.

rates.

6.4.3 Discussion

We observe a decline in performance compared with classification via LDA; we see a reduction of approximately 7% in each our performance measures which suggests that the complex super-topic/topic structure is inappropriate for the modelling of text in mathematical documents, especially when compared to the successes of LDA.

One possible reason for the decline in performance could be due to the inference methods used. In particular, we introduce Online Variational Bayes for Pachinko Allocation which allows us to perform extensive numbers of experiments quickly. Although we may have sacrificed performance efficiency by using such

methods, the choice of using Online Variational Bayes methods is not necessarily a problem concerning our research goals; the aims of this thesis are to examine the effect on mathematical document classification performance when modelling mathematical documents over dual vocabularies. We keep the inference techniques of our models as similar as possible to ensure that we are only testing the impact of the mathematical symbols on classification performance, and not the strength of the inference techniques. Furthermore, Pachinko Allocation is not directly applicable to our problem of mathematical document classification since PA assumes observations over a single vocabulary.

To summarise, we have introduced PA as a stepping stone to the more powerful Dual Pachinko Allocation document model which assumes observations over a dual vocabulary. We describe Dual Pachinko Allocation in detail in Chapter 7.

Chapter 7

Dual Pachinko Allocation

We now introduce our final mathematical document model *Dual Pachinko Allocation*. In the previous chapter, we introduce Pachinko Allocation, which captures correlations between words into topics and furthermore correlations between topics into super-topics. We now generalise PA to develop *Dual Pachinko Allocation*, a novel probabilistic generative model which assumes observations over dual vocabularies.

We recall our experimental results of classification via DLDA: we observe a performance decrease when compared with classification via the equivalent single-vocabulary LDA model. We attribute the decline in performance to the possibly incorrect assumptions made under the DLDA model; that there is a one-to-one correspondence between the word and symbol topics. We postulate that this is not the case since there are likely instances of mathematical subject areas, especially in the areas of applied sciences, which share notation but not terminology.

The novelty of Dual Pachinko Allocation is that it allows for topics over each vocabulary to be modelled separately which in turn relaxes the assumption that there is a one-to-one correspondence between the word topics and symbol topics.

The Dual Pachinko Allocation model extends the Pachinko Allocation model a similar way to how DLDA extends LDA. In the context of mathematical document modelling, Dual Pachinko Allocation assumes each document can be represented as a mixture of *two* collections of latent topics: a mixture of word topics and symbol topics. In particular, the mixtures of word topics and mixtures of symbol topics are conditionally independent given the super-topic mixture. That

is, for a corpus of mathematical documents, the correlations between words and correlations between symbols are not directly influenced by one another, but instead influenced by a mixture of latent super-topics. As before, the word topics and symbol topics are each characterised by distributions of words and symbols respectively, but under the assumptions of Dual Pachinko Allocation, there may now be different numbers of word topics and symbol topics.

To summarise, in this chapter, we first provide a detailed outline of the Dual Pachinko Allocation model, introduce Batch and Online Variational Bayes algorithms for parameter estimation, describe mathematical document classification via Dual Pachinko Allocation, and finally, perform various sets of experiments and discuss classification on mathematical corpora.

7.1 Dual Pachinko Allocation

The Dual Pachinko Allocation model (DPA) extends Pachinko Allocation to model collections of discrete data where observations span dual vocabularies. In the context of mathematical document modelling, DPA assumes mathematical documents as collections words and symbols, which can be represented by a collection of mixtures of latent topics. In particular, mixtures of latent word topics $\boldsymbol{\theta}_{di}$, and mixtures of latent symbol topics $\boldsymbol{\theta}^s_{di}$, where the influence of each topic mixture is weighted according to a mixture of latent super-topics $\boldsymbol{\theta}^r_d$. Similar to DLDA, a word and symbol topics are characterised by distributions of words and symbols respectively.

The generative process is similar to DLDA and PA. DPA assumes that a document can be generated by first sampling a latent super-topic mixture and a collection of word topic and symbol topic mixtures, then sampling a collection of words and symbols according to the weights in the super-topic and word/symbol topic mixtures.

Formally, this model assumes that the number of latent super-topics S, the numbers of the latent word and symbols topics K and K^s, and the sizes of the word and symbol vocabularies V and V^s are known and fixed. Furthermore, the numbers of words N_d and symbols N^s_d are Poisson distributed with parameters ξ and ξ^s respectively. Finally, the K word topics and K^s symbol topics are Dirichlet distributed with smoothing parameters $\boldsymbol{\eta}$ and $\boldsymbol{\eta}^s$ respectively. Similar to DLDA,

the Poisson assumptions here are not critical for this Chapter, but the fact that the numbers of words and symbols belong to different distributions is crucial.

Given the Dirichlet priors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^s$ on the word and symbol topic mixtures, and $\boldsymbol{\alpha}^r$ on the super-topic mixtures; word topics $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$; and symbol topics $\boldsymbol{\beta}^s_1, \ldots, \boldsymbol{\beta}^s_{K^s}$; DPA assumes the generative process of a document $(\mathbf{w}_d, \mathbf{s}_d)$ outlined in Algorithm 13 where we make clear the similarities to the PA generative process by highlighting the new steps introduced.

Algorithm 13 Generative process of a document under DPA

Sample super-topic mixture $\boldsymbol{\theta}^r \sim \text{Dir}(\boldsymbol{\alpha}^r)$
for each of the S super-topics **do**
  Sample word topic mixture $\boldsymbol{\theta}_{di} \sim \text{Dir}(\boldsymbol{\alpha}_i)$
  Sample symbol topic mixture $\boldsymbol{\theta}^s_{di} \sim \text{Dir}(\boldsymbol{\alpha}^s_i)$ ▷ New Step
end for
Sample number of words $N_d \sim \text{Poisson}(\xi)$
for each of the N_d words **do**
  Sample word super-topic index $i = z_{dn} \sim \text{Cat}(\boldsymbol{\theta}^r_d)$
  Sample word topic index $j = z'_{dn} \sim \text{Cat}(\boldsymbol{\theta}_{di})$
  Sample word index $\mathbf{w}_{dn} \sim \text{Cat}(\boldsymbol{\beta}_j)$
end for
Sample number of symbols $N^s_d \sim \text{Poisson}(\xi^s)$
for each of the N^s_d symbols **do** ▷ New Step
  Sample symbol super-topic index $i = z^s_{dn} \sim \text{Cat}(\boldsymbol{\theta}^r_d)$ ▷ New Step
  Sample symbol topic index $j = z^{s\prime}_{dn} \sim \text{Cat}(\boldsymbol{\theta}^s_{di})$ ▷ New Step
  Sample symbol index $\mathbf{s}_{dn} \sim \text{Cat}(\boldsymbol{\beta}^s_j)$ ▷ New Step
end for

Figures 7-1 and 7-2 show exampled of the super-topic, word topic mixtures and symbol topic mixtures for three documents and three word and symbol topics under the DPA model.

The similarity between the generative processes of DPA and PA is similar to the similarities between DLDA and LDA. In particular, the main difference is the introduction of K^s symbol topics $\boldsymbol{\beta}^s_j$ and for each observation, we also generate N^s_d symbols with corresponding super-topic and symbol topic indices. Unlike DLDA, we introduce possibly a different number of symbol topics to the number

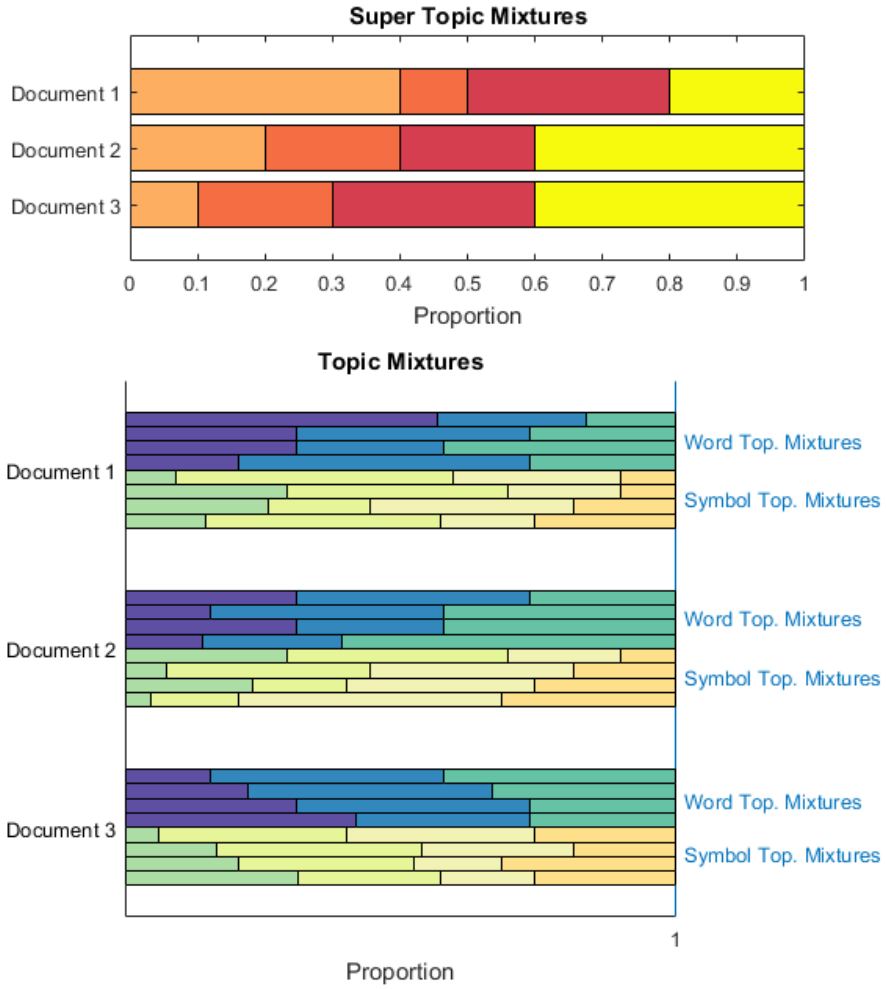

Figure 7-1: Example super-topic, word topic, and symbol topic mixtures under DPA.

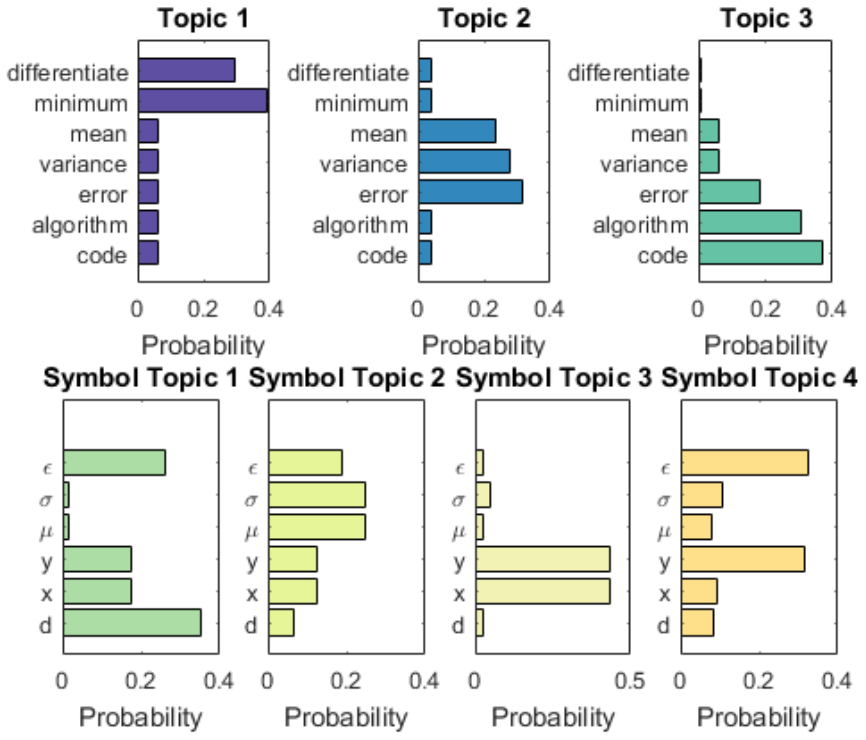

Figure 7-2: Example word topics and symbol topics under DPA.

of word topics.

Under the generative process of DPA, given the model parameters, the joint distribution of a document $(\mathbf{w}_d, \mathbf{s}_d)$ with super-topic mixture $\boldsymbol{\theta}^r_d$; topic mixtures $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}^s_d$; super-topic indices $\mathbf{z}_d$ and $\mathbf{z}^s_d$; topic indices $\mathbf{z}'_d$ and $\mathbf{z}'^s_d$; and topics $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^s$ is given by the product

$$\begin{aligned} p(\mathcal{H}, \mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta}) = p_{\text{PA}}(\mathcal{H}, \mathbf{w}_d|\boldsymbol{\Theta}) \prod_{i=1}^{S} p(\boldsymbol{\theta}^s_{di}|\boldsymbol{\alpha}^s_i) \prod_{j=1}^{K^s} p(\boldsymbol{\beta}^s_j|\boldsymbol{\eta}^s) \\ \times \prod_{n=1}^{N^s_d} p(\mathbf{z}^s_{dn}|\boldsymbol{\theta}^r_d) p(\mathbf{z}'^s_{dn}|\mathbf{z}^s_{dn}, \boldsymbol{\theta}^s_d) p(\mathbf{s}_{dn}|\mathbf{z}'^s_{dn}, \boldsymbol{\beta}^s) \end{aligned} \tag{7.1}$$

where $\mathcal{H}$ denotes the set of latent variables, $\boldsymbol{\Theta}$ denotes the set of model parameters, and p_{PA} denotes the joint distribution of a document under PA given by Equation (6.1). The symbol specific factors $p(\boldsymbol{\theta}^s_{dn}|\boldsymbol{\alpha}^s_i)$ and $p(\boldsymbol{\beta}^s_j|\boldsymbol{\eta}^s)$ are given by the probability density function of the Dirichlet distribution; and the remaining factors are given by the probability denstiy function of the Categorical distribu-

tion where $p(\mathbf{z'}^s_{dn}|z^s_{dn}=i,\boldsymbol{\theta}^s_d)$ and $p(\mathbf{s}_{dn}|z'^s_{dn}=j,\boldsymbol{\beta}^s)$ are given by $p(\mathbf{z'}^s_{dn}|\boldsymbol{\theta}^s_{di})$ and $p(\mathbf{s}_{dn}|\boldsymbol{\beta}^s_j)$ respectively.

Figure 7-3 describes the DPA model as a probabilistic graphical model. The graphical model makes clear the similarities to PA in Figure 6-2 and DLDA in Figure7-4. In particular, DPA has the same four-level structure as PA and furthermore, similar to DLDA, removing the symbol specific nodes yields the graphical model for PA due to the conditional independence of the word and symbol branches given the super-topic mixtures. Similar to PA, if we consider the case when $S = 1$, the super-topic specific variables become trivial, and we obtain the DLDA graphical model.

7.2 Inference

In this section, we outline the process for solving for latent variables in the DPA model. In particular, given the model parameters, we wish to determine the latent variables which maximise the likelihood of a document $(\mathbf{w}_d, \mathbf{s}_d)$ via the posterior distribution

$$p(\mathcal{H}|\mathbf{w}_d, \mathbf{s}_d, \boldsymbol{\Theta}) = \frac{p(\mathcal{H}, \mathbf{w}_d, \mathbf{s}_d\ \boldsymbol{\Theta})}{p(\mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta})}$$

Again, this posterior distribution is intractable to compute in general due to the interaction between the topic mixtures and the topics. We now outline the Batch Variational Bayes inference procedure for DPA.

7.2.1 Variational Inference

Recall that the strategy for Variational Bayes is to obtain the tightest lower bound on the log-likelihood by optimising this lower bound over a set of free variational parameters. We construct this lower bound using the same strategy as before. We consider a simpler version of the graphical model for DPA with the model parameters, edges and observed variables removed and augment this model with a set of free variational parameters. In particular, we augment the model with the same variational parameters as in PA as well as the symbol specific equivalents

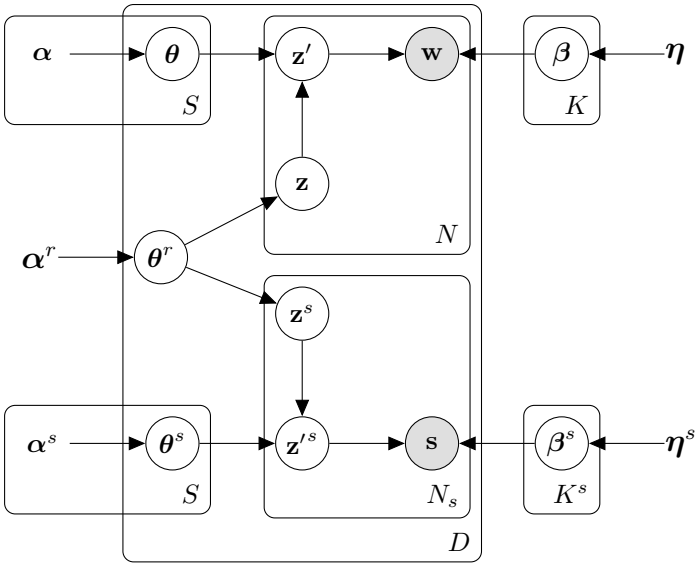

Figure 7-3: Graphical model representation of DPA

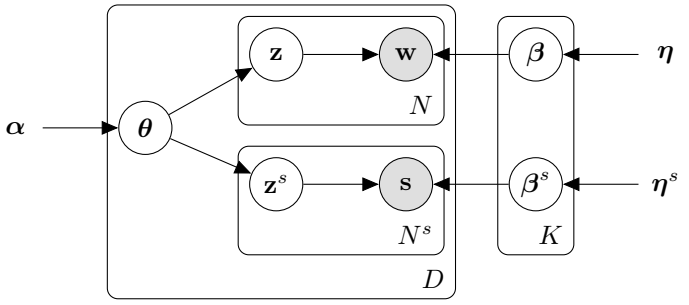

Figure 7-4: Graphical model representation of DLDA

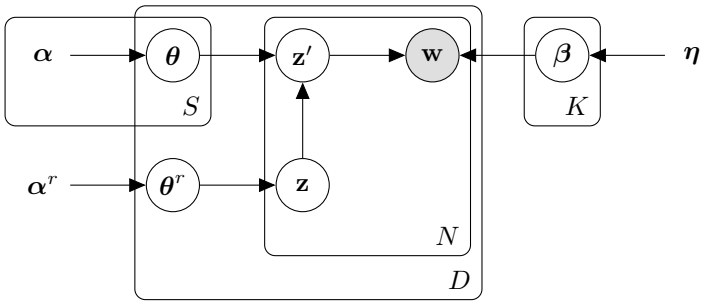

Figure 7-5: Graphical model representation of PA

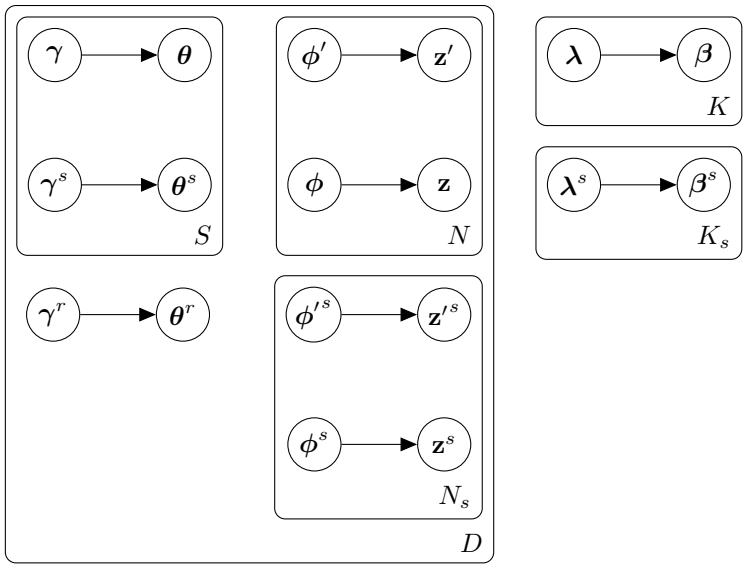

Figure 7-6: Graphical model representation of the variational distribution of DPA

$\boldsymbol{\gamma}^s$, $\boldsymbol{\phi}^s$, $\boldsymbol{\phi}'^s$, and $\boldsymbol{\lambda}^s$ with dependencies characterised by the variational given by

$$q(\mathcal{H}|\mathcal{F}) = \prod_{j=1}^{K} q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j) \prod_{j=1}^{K^s} q(\boldsymbol{\beta}_j^s|\boldsymbol{\lambda}_j^s) \prod_{d=1}^{D} q_d(\mathcal{H}|\mathcal{F}) \tag{7.2}$$

were $\mathcal{F}$ denotes the set of free variational parameters, and q_d denotes the variational distribution of a document given by

$$\begin{aligned} q_d(\mathcal{H}|\mathcal{F}) = p(\boldsymbol{\theta}_d^r|\boldsymbol{\gamma}_d^r) \prod_{i=1}^{S} q(\boldsymbol{\theta}_{di}|\boldsymbol{\gamma}_{di}) \prod_{n=1}^{N_d} q(\mathbf{z}'_{dn}|\boldsymbol{\phi}'_{dn}) q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn}) \\ \times \prod_{n=1}^{N_d^s} q(\mathbf{z}'^s_{dn}|\boldsymbol{\phi}'^s_{dn}) q(\mathbf{z}^s_{dn}|\boldsymbol{\phi}^s_{dn}) \end{aligned} \tag{7.3}$$

The factors of Equations (7.2) and (7.3) which also appear in variational distribution of the PA model are given by the same probability distributions as before. The remaining symbol specific factors $q(\boldsymbol{\beta}_j^s|\boldsymbol{\lambda}_j^s)$ and $q(\boldsymbol{\theta}_{di}^s|\boldsymbol{\gamma}_{di}^s)$ are given by the probability density function of the Dirichlet distribution; and $q(\mathbf{z}_{dn}^s|\boldsymbol{\phi}_{dn}^s)$ and $q(\mathbf{z}'^s_{dn}|\boldsymbol{\phi}'^s_{dn})$ are given by the probability density function of the Categorical distribution. Figure 7-6 describes the full variational distribution as a probabilistic graphical model.

We obtain a lower bound of the log-likelihood using Jensen's inequality via the same method as deriving the lower bound for the log-likelihood of a document under DLDA. Furthermore, we arrive at the same lower bound as a function of expectations over the variational distribution q given by the inequality

$$\log p(\mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta}) \geq \mathbb{E}_q[\log p(\mathcal{H}, \mathbf{w}_d, \mathbf{s}_d|\boldsymbol{\Theta})] - \mathbb{E}_q[\log q_d(\mathcal{H}|\mathcal{F})] \tag{7.4}$$

We let $\mathcal{L}$ denote the right hand side of Equation (7.4) as a function of the free variational parameters $\mathcal{F}$ given the model parameters $\boldsymbol{\Theta}$ and call this the *Evidence Lower Bound*. We rewrite $\mathcal{L}$ using the factorisations of p and q as

$$\begin{aligned}\mathcal{L}(\mathcal{F}; \boldsymbol{\Theta}) = \sum_{d=1}^{D} L_d(\mathcal{F}; \boldsymbol{\Theta}) + \mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] + \mathbb{E}_q[\log p(\boldsymbol{\beta}^s|\boldsymbol{\eta}^s)] \\ - \mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] - \mathbb{E}_q[\log q(\boldsymbol{\beta}^s|\boldsymbol{\lambda})]\end{aligned} \tag{7.5}$$

where L_d denotes the contribution of the dth document to the Evidence Lower Bound. In particular, due to the similarities between DPA and PA, we discover that this contribution is the same as the equivalent contribution under PA, but with the addition of the symbol specific expectations

$$\begin{aligned}L_d(\mathcal{F}; \boldsymbol{\Theta}) = L_d^{\mathrm{PA}}(\mathcal{F}; \boldsymbol{\Theta}) + \mathbb{E}_q[\log p(\boldsymbol{\theta}_d^s|\boldsymbol{\alpha}^s)] + \mathbb{E}_q[\log p(\mathbf{z}_d^s|\boldsymbol{\theta}_d^r)] \\ + \mathbb{E}_q\big[\log p(\mathbf{z'}_d^s|\mathbf{z}^s, \boldsymbol{\theta}_d^s)\big] + \mathbb{E}_q[\log p(\mathbf{s}_d|\mathbf{z'}_d^s, \boldsymbol{\beta}^s)] - \mathbb{E}_q[\log q(\boldsymbol{\theta}_d^s|\boldsymbol{\gamma}_d^s)] \\ - \mathbb{E}_q[\log q(\mathbf{z}_d^s|\boldsymbol{\phi}_d^s)] - \mathbb{E}_q\big[\log q(\mathbf{z'}_d^s|\boldsymbol{\phi'}_d^s)\big]\end{aligned} \tag{7.6}$$

where L_d^{PA} corresponds to the equivalent contribution of the dth document under PA given by Equation (6.6).

We now have a lower bound on the log-likelihood of a corpus given an arbitrary variational distribution q. Similar to before, we wish to minimise the difference between the log-likelihood and the Evidence Lower Bound where we note that this difference is the KL divergence between the posterior probability $q(\mathcal{H}|\mathcal{F})$ and the true posterior probability $p(\mathcal{H}|\mathbf{w}, \boldsymbol{\Theta})$. We now express the log-likelihood in terms of the Evidence Lower Bound and this KL divergence:

$$\log p(\mathbf{w}, \mathbf{s}|\boldsymbol{\Theta}) = \mathcal{L}(\mathcal{F}; \boldsymbol{\Theta}) + \mathrm{D}(q(\mathcal{H}|\mathcal{F}) \,\|\, p(\mathcal{H}|\mathbf{w}, \mathbf{s}, \boldsymbol{\Theta})$$

Here we see that minimising the KL divergence between the variational and true posterior probabilities is equivalent to maximising $\mathcal{L}$ via the free variational parameters. We now outline the steps for maximising the Evidence Lower Bound.

We express each of the expectations in Equations (7.5) and (7.6) as their expanded forms outlined in Appendix B. Furthermore, we discover that the expectations over the topics $\boldsymbol{\beta}^*$ have identical expansions to the DLDA case, and those residing in L_d^{PA} have identical expansions to the PA case. The remaining expectations in Equation (7.5) expand as follows.

Appendices B.2 and B.3 show the expectations over the symbol topic mixtures $\boldsymbol{\theta}_d^s$ expand to

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d^s|\boldsymbol{\alpha}^s)] = \sum_{i=1}^{S}\left(\log\Gamma\left(\sum_j \alpha_{ij}^s\right) + \sum_{j=1}^{K^s}\left((\alpha_{ij}^s - 1)\mathbb{E}_q\left[\log\theta_{dij}^s|\boldsymbol{\gamma}_{di}^s\right] - \log\Gamma\left(\alpha_{ij}^s\right)\right)\right)$$
$$\mathbb{E}_q[\log q(\boldsymbol{\theta}_d^s|\boldsymbol{\alpha}^s)] = \sum_{i=1}^{S}\left(\log\Gamma\left(\sum_j \gamma_{ij}^s\right) + \sum_{j=1}^{K^s}\left((\gamma_{ij}^s - 1)\mathbb{E}_q\left[\log\theta_{dij}^s|\boldsymbol{\gamma}_{di}^s\right] - \log\Gamma\left(\gamma_{ij}^s\right)\right)\right)$$

where we emphasis the dependency on $\boldsymbol{\gamma}_d^s$ in the inner expectations. Appendices B.4 and B.5 show the expectations over the symbol super-topic indices $\mathbf{z}_d^s$ expand and rearrange to give the summations

$$\mathbb{E}_q[\log p(\mathbf{z}_d^s|\boldsymbol{\theta}_d^r)] = \sum_{n=1}^{N_d^s}\sum_{i=1}^{S}\phi_{dni}^s\mathbb{E}_q[\log\theta_{di}^r|\boldsymbol{\gamma}_d^r]$$
$$\mathbb{E}_q[\log q(\mathbf{z}_d^s|\boldsymbol{\phi}_d^s)] = \sum_{n=1}^{N_d^s}\sum_{i=1}^{S}\phi_{dni}^s\log\phi_{dni}$$

where we emphasis the dependency on $\boldsymbol{\gamma}_d^r$ in the inner expectations. Similarly, Appendix B.5 also shows that

$$\mathbb{E}_q\left[\log q(\mathbf{z'}_d^s|\boldsymbol{\phi'}_d^s)\right] = \sum_{n=1}^{N_d^s}\sum_{j=1}^{K}\phi'^s_{dnj}\log\phi'^s_{dnj}$$

Finally, Appendix B.6 shows the remaining expectation over the symbol topic

indices ${\mathbf{z}'}^s_d$ expands and rearranges to the summation

$$\mathbb{E}_q\big[\log p({\mathbf{z}'}^s_d|\mathbf{z}^s_d, \boldsymbol{\theta}^s_d)\big] = \sum_{n=1}^{N^s_d}\sum_{i=1}^{S}\phi^s_{dni}\sum_{j=1}^{K^s}{\phi'}^s_{dnj}\mathbb{E}_q\big[\log\theta^s_{dij}|\boldsymbol{\gamma}^s_d\big]$$

where we emphasise the dependency on $\boldsymbol{\gamma}^s_d$ on the inner expectations.

By plugging in the above expectations into $\mathcal{L}$, we now have the Evidence Lower Bound as a function of the model parameters and the free variational parameters. We now outline the process of maximising this lower bound via the free variational parameters.

7.2.1.1 Document Level Updates

In this section, we describe the methods of maximising $\mathcal{L}$ with respect to each of the document level variational parameters $\boldsymbol{\phi}^*_d$, ${\boldsymbol{\phi}'}^*_d$, $\boldsymbol{\gamma}^*_d$ and $\boldsymbol{\gamma}^r_d$.

Variational Categorical Parameters The strategy for finding the maximising values of ϕ_{dni} and ϕ'_{dnj} under DPA is identical to finding the same maximising values under PA as described in Section 6.2.1. Furthermore, since the DPA model does not introduce any more terms to $\mathcal{L}$ which depend on ϕ_{dni} and ϕ'_{dnj}, we find that the maximising values are also identical.

Following the same strategy for finding the maximising values of ϕ^s_{dni}, we discover that the maximising values are the same form as ϕ_{dni} but with the word specific and symbol specific variables interchanged. That is, the maximising value of ϕ^s_{dni} is given by

$$\phi^s_{dni} \propto \exp\left\{\mathbb{E}_q[\log\theta^r_{di}|\boldsymbol{\gamma}^r_d] + \sum_{j=1}^{K^s}{\phi'}^s_{dnj}\mathbb{E}_q\big[\log\theta^s_{dij}|\boldsymbol{\gamma}^s_{di}\big]\right\}$$

where $\boldsymbol{\phi}^s_{dn}$ is normalised to sum to one. Similarly, the maximising value of ${\phi'}^s_{dnj}$ given by

$$ {\phi'}^s_{dnj} \propto \exp\left\{\sum_{i=1}^{S}\phi^s_{dni}\mathbb{E}_q\big[\log\theta^s_{dij}|\gamma^s_{di}\big] + \mathbb{E}_q\big[\log\beta^s_{js_{dn}}|\boldsymbol{\lambda}^s_j\big]\right\}$$

where ${\boldsymbol{\phi}'}^s_{dn}$ is normalised to sum to one.

Variational Dirichlet Parameters We now maximise the Evidence Lower Bound via the Variational Dirichlet parameter γ_i^r; the ith component of the Dirichlet parameter on the super-topic mixtures. Since the DPA model introduces new terms $\mathcal{L}$ which depend on $\boldsymbol{\gamma}_d^r$, the maximising values are similar, but not identical. Retaining only the terms of $\mathcal{L}$ containing γ_{di}^r we have

$$\mathcal{L}_{[\gamma_{di}^r]} = \sum_{i'=1}^{S} \left(\mathbb{E}_q[\log \theta_{di'}^r | \boldsymbol{\gamma}_d^r] \left(\alpha_{i'}^r - \gamma_{di'}^r + \sum_n \phi_{dni'} + \underbrace{\sum_n \phi_{dni'}^s}_{\text{New terms}} \right) \right) - \log \Gamma \left(\sum_{i'} \gamma_{di'}^r \right) + \log \Gamma(\gamma_i^r)$$

Taking partial derivatives with respect to γ_{di}^r yields

$$\begin{aligned} \frac{\partial \mathcal{L}_{[\gamma_{di}^r]}}{\partial \gamma_{di}^r} = \Psi'(\gamma_{di}^r) & \left(\alpha_i^r - \gamma_{di}^r + \sum_{n=1}^{N_d} \phi_{dni} + \sum_{n=1}^{N_d^s} \phi_{dni}^s \right) \\ & - \Psi' \left(\sum_{i'} \gamma_{di'}^r \right) \sum_{i'=1}^{S} \left(\alpha_{i'}^r - \gamma_{di'}^r + \sum_{n=1}^{N_d} \phi_{dni'} + \sum_{n=1}^{N_d^s} \phi_{dni'}^s \right) \end{aligned}$$

Setting this derivative to zero yields the maximising value of γ_{di}^r at

$$\gamma_{di}^r = \alpha_i^r + \sum_{n=1}^{N_d} \phi_{dni} + \underbrace{\sum_{n=1}^{N_d^s} \phi_{dni}^s}_{\text{New terms}}$$

Finally, following the same strategy for finding the maximising values of γ_{dij} under PA, we find the maximising values are of the same form, but with the word specific and symbol specific variables interchanged. That is, the maximising value of γ_{dij}^s is given by

$$\gamma_{dij}^s = \alpha_{ij}^s + \sum_{n=1}^{N_d^s} \phi_{dni}^s \phi'^s_{dnj}$$

We now have the update rules for the variational parameters which we require for the document level variational inference procedure. Since the update rules for the document level variational parameters are dependent on one another, full variational inference requires iterating through these update rules until convergence. We describe the document level variational inference procedure for Dual Pachinko Allocation in Algorithm 14.

Algorithm 14 Document level variational inference for Dual Pachinko Allocation

Initialise $\boldsymbol{\gamma}^r_d$, $\boldsymbol{\gamma}_d$, $\boldsymbol{\gamma}^s_d$ randomly

repeat

Set $\phi_{dni} \propto \exp\Big\{\Psi(\gamma_{di}) + \sum_j \phi'_{dnj}\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_{di}]\Big\}$

Set $\phi^s_{dni} \propto \exp\Big\{\Psi(\gamma^s_{di}) + \sum_j \phi'^s_{dnj}\mathbb{E}_q\big[\log\theta^s_{dij}|\boldsymbol{\gamma}^s_{di}\big]\Big\}$

Set $\phi'_{dnj} \propto \exp\{\sum_i \phi_{dni}\mathbb{E}_q[\log\theta_{dij}|\boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log\beta_{jw_{dn}}|\boldsymbol{\lambda}_j]\}$

Set $\phi'^s_{dnj} \propto \exp\big\{\sum_i \phi^s_{dni}\mathbb{E}_q\big[\log\theta^s_{dij}|\boldsymbol{\gamma}^s_{di}\big] + \mathbb{E}_q\big[\log\beta^s_{js_{dn}}|\boldsymbol{\lambda}^s_j\big]\big\}$

Set $\gamma^r_{di} = \alpha^r_i + \sum_n \phi_{dni} + \sum_n \phi^s_{dni}$

Set $\gamma_{dij} = \alpha_{ij} + \sum_n \phi_{dni}\phi'_{dnj}$

Set $\gamma^s_{dij} = \alpha^s_{ij} + \sum_n \phi^s_{dni}\phi'^s_{dnj}$

until Convergence of $\boldsymbol{\gamma}^r_d$, $\boldsymbol{\gamma}_d$, γ^s_d

Similar to the previous models, we may use the document level variational procedure to approximate the topic mixture representation of a document under DPA. We use Algorithm 14 as a function of a document $(\mathbf{w}_d, \mathbf{s}_d)$ to provide the optimised values of $\boldsymbol{\phi}'_d$ and $\boldsymbol{\phi}'^s_d$. We approximate the topic mixture representation as the expected value of the normalised frequency count of the topic indices $\mathbf{z}'_{dn}$ and $\mathbf{z}'^s_{dn}$ under the variational distribution q:

$$\bar{\boldsymbol{\phi}}'_d = \mathbb{E}_q\left[\frac{1}{N_d + N^s_d}\left(\sum_{n=1}^{N_d}\mathbf{z}'_{dn} + \sum_{n=1}^{N^s_d}\mathbf{z}'^s_{dn}\right)\right] = \frac{1}{N_d + N^s_d}\left(\sum_{n=1}^{N_d}\boldsymbol{\phi}'_{dn} + \sum_{n=1}^{N^s_d}\boldsymbol{\phi}'^s_{dn}\right)$$

where we use the fact that under the variational distribution q, the expected values of $\mathbf{z}'_{dn}$ and $\mathbf{z}'^s_{dn}$ are given by $\boldsymbol{\phi}'_{dn}$ and $\boldsymbol{\phi}'^s_{dn}$ respectively.

7.2.1.2 Corpus Level Updates

We now have a variational inference procedure for approximating the document level variational parameters under DPA. We now derive the full variational inference method for approximating the corpus level variational parameters on the topics $\boldsymbol{\lambda}^*$ and the Dirichlet priors on the topic mixtures $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}^*$.

We use the same strategy as before; we have already shown that there is a tractable lower bound on the log-likelihood, and we can further maximise this lower bound via the corpus level parameters. Thus, we can derive a full variational EM procedure that yields the optimised values for $\boldsymbol{\alpha}^r$, $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^s$, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$.

We optimise for $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$ using the same method for optimising $\boldsymbol{\lambda}$ under PA. Furthermore, we discover that the update rule for $\boldsymbol{\lambda}$ is identical, and the update rule for $\boldsymbol{\lambda}^s$ is of the same form, but with the word-specific and symbol specific variables interchanged. That is, the maximising value of $\boldsymbol{\lambda}^s_{jv}$ is given by

$$\lambda^s_{jv} = \eta^s_v + \sum_{d=1}^{D} \sum_{n=1}^{N^s_d} s_{dnv} \phi'^s_{dnj}$$

Similarly, we optimise $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}$ using the same methods as before, and in turn, obtain the same update rules as in PA. Finally, optimising $\boldsymbol{\alpha}^s$ using the same methods as in PA, we obtain update rules of the same for as those for $\boldsymbol{\alpha}$, but with the word-specific and symbol-specific variables exchanged; we optimise $\boldsymbol{\alpha}^s_i$ using [36] with the update rule

$$\boldsymbol{\alpha}^s_i \leftarrow \boldsymbol{\alpha}^s_i - \tilde{\boldsymbol{\alpha}}^s_i(\boldsymbol{\gamma}^s)$$

where $\tilde{\boldsymbol{\alpha}}^s$, as a function of $\boldsymbol{\gamma}^s$, is the inverse of the Hessian H_i multiplied by the gradient $\mathbf{g}_i$ given by

$$g_{ij} = D\left(\Psi\left(\sum_{j'} \alpha^s_{ij'}\right) - \Psi\left(\alpha^s_{ij}\right)\right) + \sum_{d=1}^{D} \mathbb{E}_q\left[\log \theta^s_{dij} | \boldsymbol{\gamma}^s_{di}\right]$$

$$H_{ijj'} = \delta(j, j') D \Psi'\left(\alpha^s_{ij}\right) - \Psi'\left(\sum_{j''} \alpha^s_{ij''}\right)$$

We now have the required document and corpus level updates necessary for the full Variational Bayes inference algorithm on the Dual Pachinko Allocation model. The outline of the variational EM procedure is as follows:

- E-step: For each document, find the optimised values of the variational parameters $\boldsymbol{\gamma}^r_d$, $\boldsymbol{\gamma}_d$, $\boldsymbol{\gamma}^s_d$, $\boldsymbol{\phi}_d$, $\boldsymbol{\phi}^s_d$, $\boldsymbol{\phi}'_d$ and $\boldsymbol{\phi}'^s_d$ using the document level variational algorithm.

- M-step: Maximise the resulting lower bound on the log-likelihood via the parameters $\boldsymbol{\alpha}^r$, $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^s$, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$ using the corpus level updates.

We summarise the procedure for full Variational Bayes inference on a corpus of mathematical documents in Algorithm 15. We apply the document level vari-

ational procedure for each document in the corpus, and update the corpus level parameters after each pass of the data.

Algorithm 15 Batch Variational Bayes for Dual Pachinko Allocation

Initialise $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^s$ randomly
repeat
  for $d = 1, \ldots, D$ **do**
    Initialise $\boldsymbol{\gamma}^r_d$, $\boldsymbol{\gamma}_d$, $\boldsymbol{\gamma}^s_d$ randomly
    repeat
      Set $\phi_{dni} \propto \exp\Big\{\Psi(\gamma_{di}) + \sum_j \phi'_{dnj} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}]\Big\}$
      Set $\phi^s_{dni} \propto \exp\Big\{\Psi(\gamma^s_{di}) + \sum_j \phi'^s_{dnj} \mathbb{E}_q\big[\log \theta^s_{dij} | \boldsymbol{\gamma}^s_{di}\big]\Big\}$
      Set $\phi'_{dnj} \propto \exp\{\sum_i \phi_{dni} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}} | \boldsymbol{\lambda}_j]\}$
      Set $\phi'^s_{dnj} \propto \exp\big\{\sum_i \phi^s_{dni} \mathbb{E}_q\big[\log \theta^s_{dij} | \boldsymbol{\gamma}^s_{di}\big] + \mathbb{E}_q\big[\log \beta^s_{js_{dn}} | \boldsymbol{\lambda}^s_j\big]\big\}$
      Set $\gamma^r_{di} = \alpha^r_i + \sum_n \phi_{dni} + \sum_n \phi^s_{dni}$
      Set $\gamma_{dij} = \alpha_{ij} + \sum_n \phi_{dni} \phi'_{dnj}$
      Set $\gamma^s_{dij} = \alpha^s_{ij} + \sum_n \phi^s_{dni} \phi'^s_{dnj}$
    until Convergence of $\boldsymbol{\gamma}^r_d$, $\boldsymbol{\gamma}_d$, γ^s_d
  end for
  Set $\lambda_{jv} = \eta_v + \sum_d \sum_n w_{dnv} \phi'_{dnj}$
  Set $\lambda^s_{jv} = \eta^s_v + \sum_d \sum_n s_{dnv} \phi'^s_{dnj}$
until Convergence of $\mathcal{L}$

7.2.2 Online Variational Inference

Algorithm 15 is a *batch* Variational Bayes algorithm; it requires a full pass of the corpus at each iteration. We now present Online Variational Bayes for Dual Pachinko Allocation, a novel extension to Batch Variational Bayes for PA which requires only a single pass of the corpus. In particular, we adapt the Batch Variational Bayes algorithm so that we update the variational parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$ at each iteration using a weighted average of their previous values and their optimal values according to the current values of $\boldsymbol{\phi}'$ and $\boldsymbol{\phi}'^s$ respectively. In particular, we update $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^s$ via the weighting parameter $\rho_d = (\tau_0 + d)^{-\kappa}$, where τ_0 denotes the offset and κ denotes the decay as before. We outline the Online Variational Bayes algorithm for Dual Pachinko Allocation in Algorithm 16.

Algorithm 16 Online variational Bayes for Dual Pachinko Allocation

Define $\rho_d := (\tau_0 + d)^{-\kappa}$
Initialise $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^s$ randomly
for $d = 1, 2, \ldots$ **do**
  Initialise γ_d^r, γ_d, $\boldsymbol{\gamma}_d^s$ randomly
  repeat
    Set $\phi_{dni} \propto \exp\left\{\Psi(\gamma_{di}^r) + \sum_j {\phi'}_{dnj} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}]\right\}$
    Set $\phi_{dni}^s \propto \exp\left\{\Psi(\gamma_{di}^r) + \sum_j {\phi'}_{dnj}^s \mathbb{E}_q\left[\log \theta_{dij}^s | \boldsymbol{\gamma}_{di}^s\right]\right\}$
    Set ${\phi'}_{dnj} \propto \exp\{\sum_i {\phi'}_{dni} \mathbb{E}_q[\log \theta_{dij} | \boldsymbol{\gamma}_{di}] + \mathbb{E}_q[\log \beta_{jw_{dn}} | \boldsymbol{\lambda}_j]\}$
    Set ${\phi'}_{dnj}^s \propto \exp\{\sum_i {\phi'}_{dni}^s \mathbb{E}_q\left[\log \theta_{dij}^s | \boldsymbol{\gamma}_{di}^s\right] + \mathbb{E}_q\left[\log \beta_{js_{dn}}^s | \boldsymbol{\lambda}_j^s\right]\}$
    Set $\gamma_{di}^r = \alpha_i^r + \sum_n \phi_{dni} + \sum_n \phi_{dni}^s$
    Set $\gamma_{dij} = \alpha_{ij} + \sum_n \phi_{dni} {\phi'}_{dnj}$
    Set $\gamma_{dij}^s = \alpha_{ij}^s + \sum_n \phi_{dni}^s {\phi'}_{dnj}^s$
  until Convergence of $\boldsymbol{\gamma}_d^r$, $\boldsymbol{\gamma}_d$, $\boldsymbol{\gamma}_d^s$
  Set $\tilde{\lambda}_{jv} = \eta_v + D \sum_n w_{dnv} {\phi'}_{dnj}$
  Set $\tilde{\lambda}_{jv}^s = \eta_v^s + D \sum_n s_{dnv} {\phi'}_{dnj}^s$
  Set $\boldsymbol{\lambda} = (1 - \rho_d)\boldsymbol{\lambda} - \rho_d \tilde{\boldsymbol{\lambda}}$
  Set $\boldsymbol{\lambda}^s = (1 - \rho_d)\boldsymbol{\lambda}^s - \rho_d \tilde{\boldsymbol{\lambda}}^s$
end for

7.3 Document Classification

We now outline document classification via Dual Pachinko Allocation. By design, the framework is almost identical to classification via LDA. In particular, we describe a semi-supervised classifier training on a partially labelled corpus via their topic mixture representations.

7.3.1 Framework

We outline the complete framework of a document via Dual Pachinko Allocation. As before, we break down the process into two layers: the unsupervised layer (the document modelling step) and the supervised later (the supervised training step).

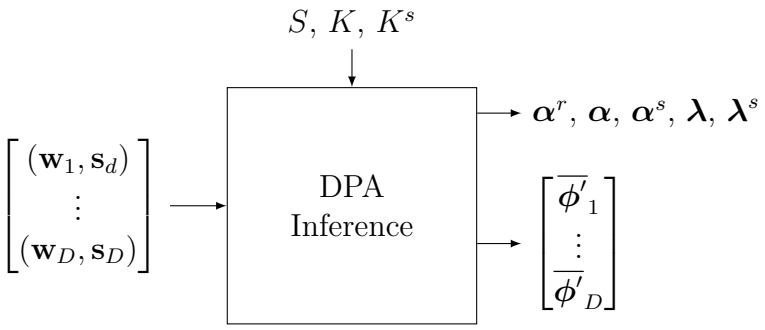

Figure 7-7: The Unsupervised Layer using DPA

Document Modelling

Given a partially labelled corpus and choices of S, K and K^s, we use the DPA variational inference methods to approximate the Dirichlet priors $\boldsymbol{\alpha}^r$ and $\boldsymbol{\alpha}^*$, the variational parameters $\boldsymbol{\lambda}^*$ on the topics, and the topic mixture representations of the documents $\bar{\boldsymbol{\phi}}'_d$. Figure 7-7 show a diagrammatic illustration of this layer.

Compared with classification via PA, the unsupervised layer for classification via DPA requires the symbol data from the documents and outputs two extra parameters: the variational parameter $\boldsymbol{\lambda}$ on the topics and the Dirichlet prior α^s on the symbol topic mixtures.

Supervised Training

The supervised layer is identical to before. Using the labelled topic mixture representations of the document obtained from the unsupervised layer as feature vectors, we train a supervised classifier. Again, we may use any discriminative supervised classification methods. We continue to use nearest neighbour methods which we describe in Appendix C.2.

Classification

We now have all the tools necessary for classification via DPA. The unsupervised layer provides the required parameters to obtain the topic mixture representation $\bar{\boldsymbol{\phi}}'$ of an unseen document. The supervised layer provides the document classifier f which will output the predicted set of labels $\mathbf{c}$ for this document. Figure 7-8 outlines the classification process of a previously unseen document.

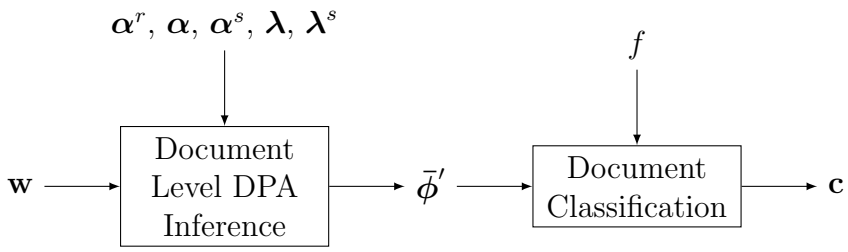

Figure 7-8: Classification process via DPA

To summarise, DPA provides a fast filtering algorithm for feature selection for document classification. In particular, DPA allows us to transform a document of possibly many words and symbols to a mixture or a much smaller number of latent topics. Since DPA controls the correlations between word topics and symbol topics at the super-topic level, the approximated topics are less prone to overlap. Therefore, DPA provides a means of representing documents as a mixture of more discriminative latent topics compared with the representations according to the previous models.

7.4 Experimental Results

In this section, we evaluate the performance of mathematical document classification via DPA using the experimental set-up as described in Chapter 3. We first perform a set of preliminary experiments to get a feel of how classification via Dual Pachinko Allocation behaves in general. In particular, we observe what effect the choices of the number of latent super-topics S, and the number latent word and symbol topics K and K^s respectively has on the classification performance. After identifying the optimal choices of the S, K and K^s, we study the best performing classifiers in detail and investigate the confusion between subject areas.

7.4.1 Preliminary Experiments

Figure 7-9 shows the effect that the choice of S, K, K^s taking values in the range $\{10, 25, 50, 100, 200\}$ has on the classification performance via DPA on the multi-labelled and uni-labelled datasets. We use the same set-up for the supervised

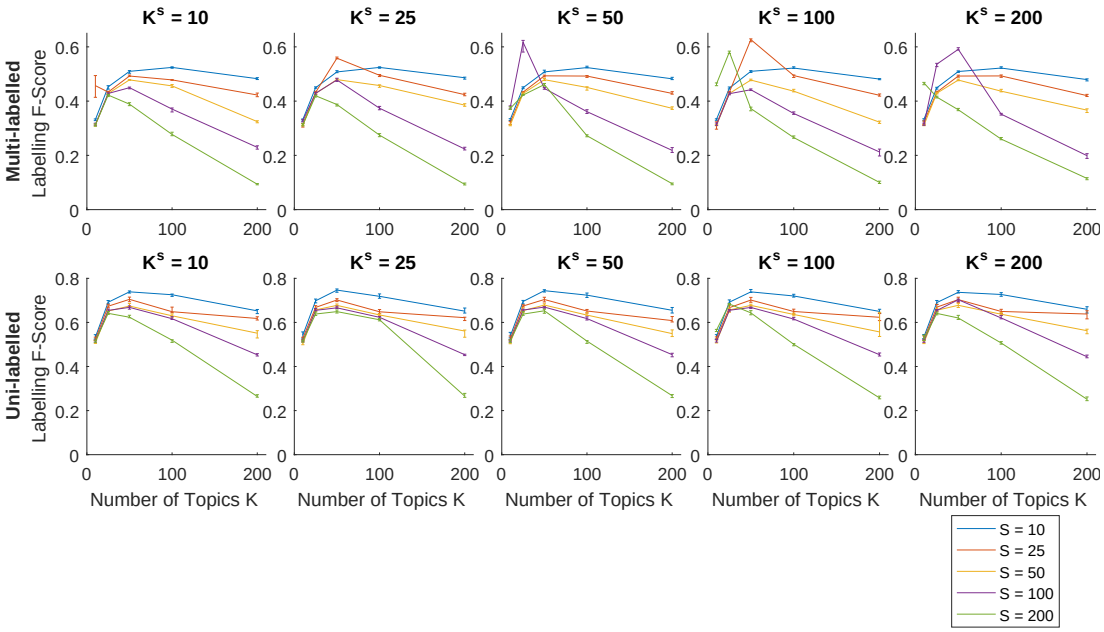

Figure 7-9: Classification performance via DPA

layer as before; we use an ensemble of Nearest Neighbour Classifiers using five nearest neighbours, inverse distance weighing and the χ^2 distance metric. Each experiment is repeated sixteen times on different random test and train partitions of the data. We plot the median and interquartile range of the Labelling F-Score on each set of experiments.

Effect of S

Firstly, looking at the effect of the choice of S, we see that in general, but with a few notable exceptions, that as S decreases, classification performance increase independently from the choices of K and K^s. This observation is similar to the single vocabulary case in the previous chapter although we do notice that we achieve the highest classification scores when $S = 25$ and $S = 100$.

Effect of K

Secondly, looking at the effect of the choice of K, we observe more obvious peaks follows by declines. We recall that in the LDA experiments, we had observed an increase in performance as K increased until levelling off around the $K = 200$

mark. In the DPA case, we see peaks in performance much earlier on; near the $K = 50$, $K = 100$ mark, followed by a decrease in performance.

Effect of K^s

Finally, looking at the effect of the choices of K^s, we notice a sharp increase in performance in some select cases; we notice some spikes in performance when $K^s \in \{50, 100, 200\}$ but this not consistent over the various combinations of values for S and K.

To summarise, we notice some fascinating interactions between the choices of S, K and K^s. The selections of S and K seem to have similar effects the earlier models in this thesis; however, we observe some interesting surges in performance in some select cases. We achieve the highest Labelling F-Score with the values $S = 10$, $K = 50$ and $K^s = 25$ on the uni-labelled dataset and $S = 100$, $K = 25$ and $K^s = 50$ on the multi-labelled dataset. For the remaining experiments, we will focus on these cases only.

Precision/Recall Trade-off

The results here look promising; we see excellent classification performance on the multi-labelled dataset. We again encounter an imbalance of precision and recall. In particular, over the sixteen experiments on each dataset, we observe a median micro-averaged precision and recall of 84.78% and 74.67% respectively on the uni-labelled classification and 85.93% and 52.06% respectively on the multi-labelled classification. Again, this implies that the classifiers are indeed too harsh when yielding positive classifications. We rerun these experiments with varying fixed thresholds and observe optimal performance when using a decision threshold of 0.3. We outline the optimal parameter settings and performance in Table 7.1 below.

7.4.2 Confusion

In this section, we investigate the per-class confusion of our classification results and furthermore, we briefly compare performance to classification via LDA. Later, in Chapter 8, we will collect the results and compare all models presented in this thesis.

	Multi-labelled	Uni-labelled
S	100	10
K	25	50
K^s	50	25
Threshold	0.3	0.3
Labelling F-Score		
(median)	69.91%	79.80%
(maximum)	70.88%	80.82%
Micro-averaged F-Score		
(median)	69.10%	79.61%
(maximum)	69.99%	80.85%

Table 7.1: Optimised Classification Performance via Online Dual Pachinko Allocation

Firstly, Figure 7-10 shows the true positive and false negative rates on the classifier on the uni-labelled dataset. The true positive and false negative rates presented here do not show anything interesting compared to the earlier models. We see a slight decline in performance regarding the range of true positive rates. We again see the strongest confusion occurs between closely related subject areas, such as between "Statistics" and "Probability theory and stochastic processes".

We now construct the equivalent confusion matrix for the classification on the multi-labelled dataset. Again, due to the size of this matrix, we only present three sections of the matrix which highlight the areas with strongest and weakest performance for readability. We present the full heat-map in Appendix D.

Firstly, Figure 7-11 shows the section of the confusion matrix highlighting the strongest per-class true positive rates; the upper-left section. Here we see an overall increase in performance compared to the earlier models. In particular, amongst these top ten performing categories we see a consistent increase in the true positive rates. We again see very slight confusion between many of the pairs of categories here.

Figure 7-12 shows the section of the confusion matrix highlighting the strongest confusion between subject areas; the lower-left section. Here we see similar levels of confusion compared to earlier models, and furthermore, we see the strongest confusion between related subject areas.

Finally, Figure 7-13 shows the section of the confusion matrix highlighting the

Figure 7-10: Per-class confusion of classification via DPA on the uni-labelled dataset

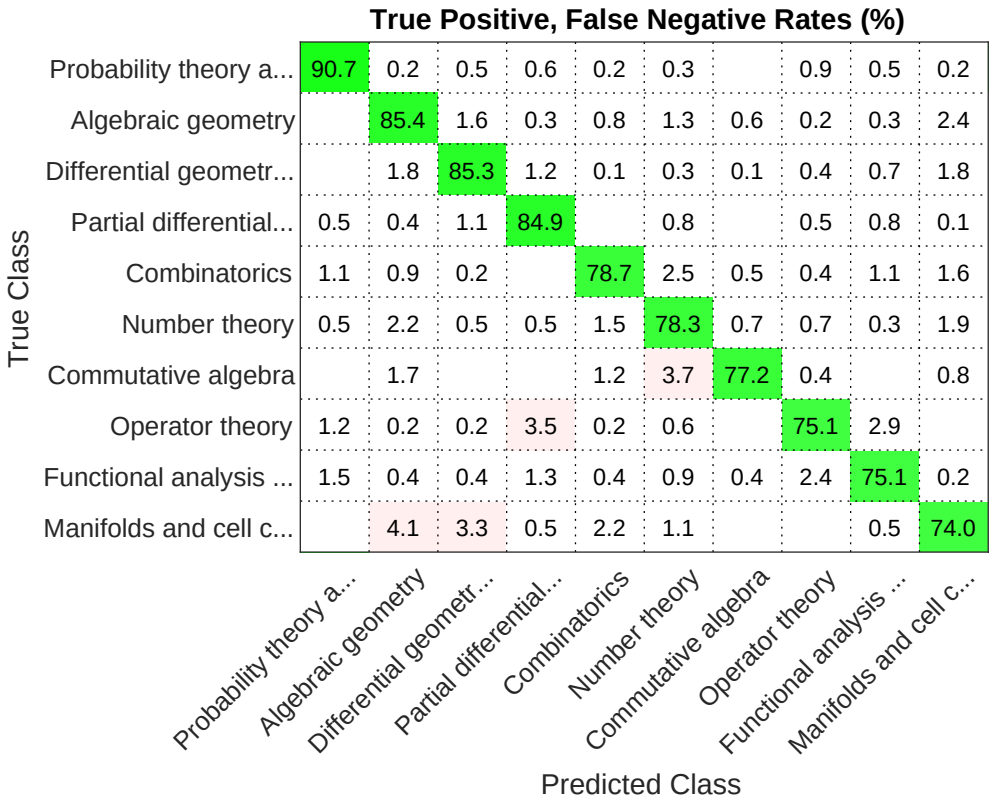

Figure 7-11: Per-class confusion of classification via DPA on the multi-labelled dataset - Upper-left section.

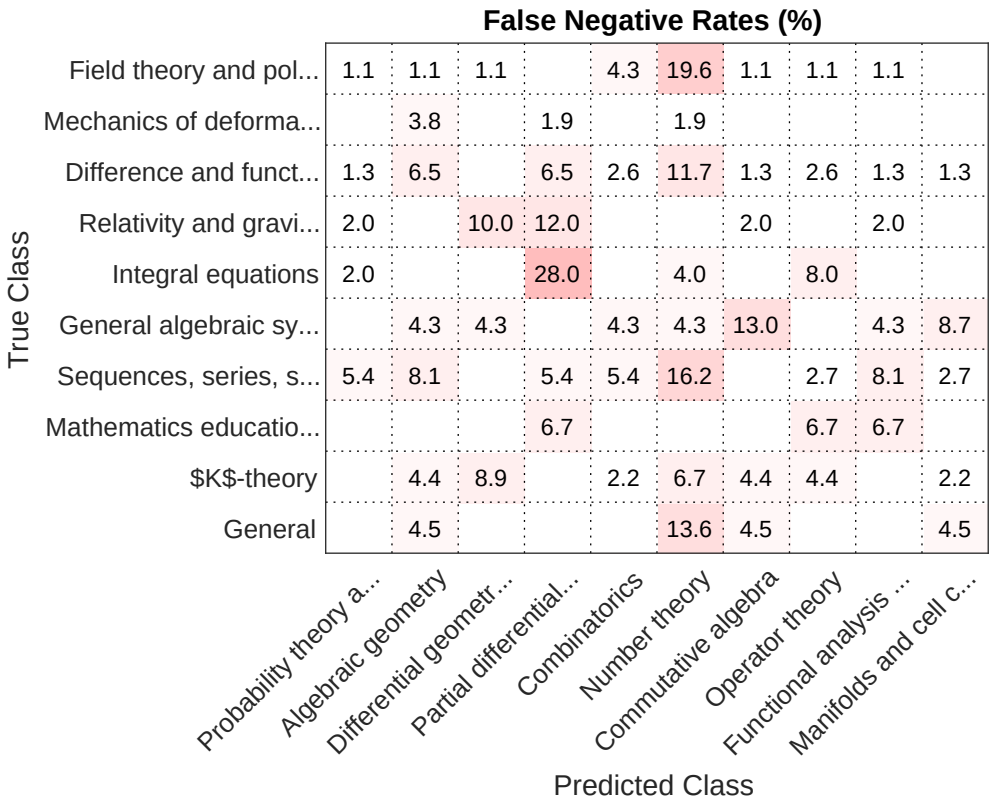

Figure 7-12: Per-class confusion of classification via DPA on the multi-labelled dataset - Lower-left section.

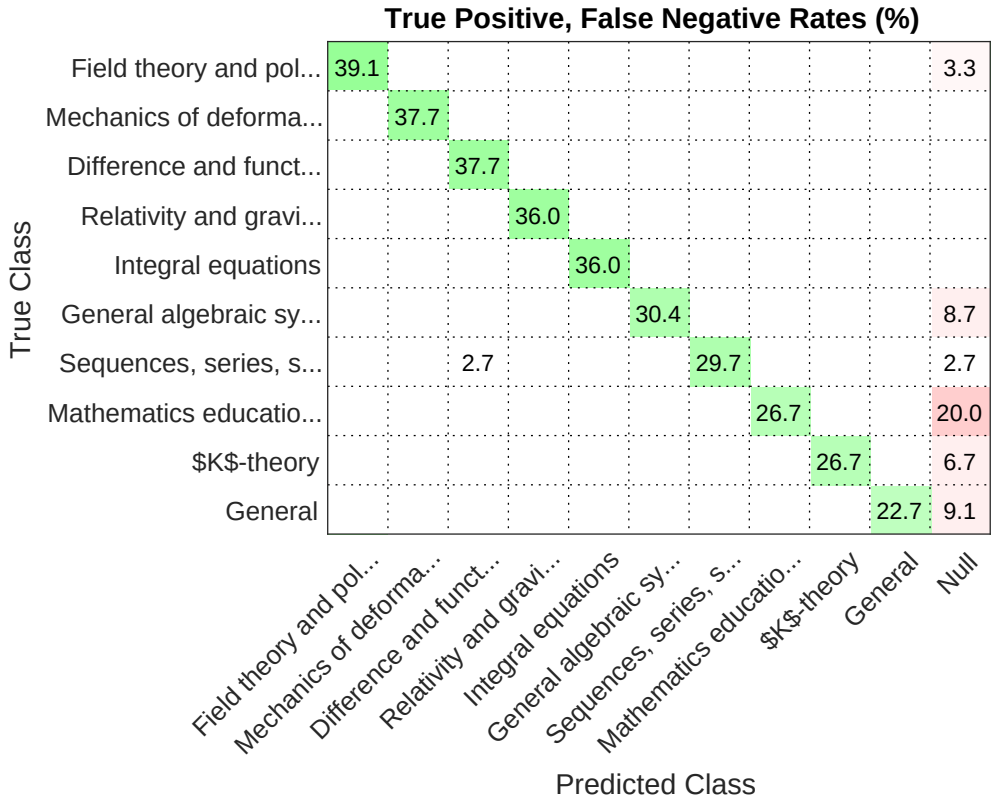

Figure 7-13: Per-class confusion of classification via LDA on the multi-labelled dataset - Lower-right section.

weakest per-class true positive rates; the lower-right section. Here we see the merits of classification via DPA. Compared to classification via LDA, we see a sharp increase in performance over the worst performing categories, with increases of true positive rates of 16%-22.7% on the ten worst performing categories. Similar to before, we again see the strongest null classification rates on the more generic subject areas.

7.4.3 Discussion

The experiments presented in this chapter have presented some interesting results. Firstly, we observe that classification via Dual Pachinko Allocation yields no performance benefit over Pachinko Allocation over the uni-labelled dataset. This observation suggests that when considering mathematical documents cover-

ing exactly one of the top twenty subject areas, that the hierarchical dual vocabulary topic/super-topic structure of Dual Pachinko Allocation is unsuitable. In particular, the imposed assumptions over LDA yield additional confusion between classes.

Secondly, we observe that classification via Dual Pachinko Allocation yields a significant performance increase over all the other models presented in this thesis on the multi-labelled dataset. The performance increase over DLDA directly supports our claim regarding the incorrect assumption made under the DLDA scenario that there is a one-to-one correspondence between word and symbol topics. In particular, when relaxing this assumption, we observe a significant increase in classification performance. This observation also reinforces our comment made in Chapter 5 about modelling documents which employ mathematical concepts from other subject areas. In particular, the approximated topics are not influenced by correlations between words and symbols directly so we can represent documents as a mixture of word and symbol topics separately.

For example, under DLDA, we may detect a symbol topic which houses the correlations between typical "partial differential equations" notation, and furthermore, we would detect co-occurring word correlations between both "partial differential equations" terminology as well as "electromagnetism" terminology. This sharing of notation will ultimately present a broad word topic which spans both collections of terminology and imposes a less discriminative set of topics to use for classification. Under DPA, this situation may resolve as detecting a single symbol topic, and two distinct and more discriminative word topics, both which can appear in documents at different proportions via the topic and super-topic mixtures.

Finally, classification via Dual Pachinko Allocation yields higher performance than the models in [13]. We have identified a mathematical document model with a strong statistical and probabilistic structure which is a direct improvement on existing classification methods.

Chapter 8

Discussion

In this chapter, we collect together all the experimental results for our document models. Firstly, we compare and contrast the classification performance via each model and discuss concerning our research aims. Finally, we outline related work in the topic modelling and classification communities and discuss future directions of research.

To recap, we have performed experiments on two different subsets of the NTCIR dataset. Firstly, we perform experiments on the multi-labelled dataset which is the full corpus of mathematical documents comprising of documents labelled with possibly multiple subject areas according to top level MSC codes. Secondly, we perform experiments on the uni-labelled dataset which is a subset of the full corpus comprising of documents only labelled with exactly one of the twenty most common top level MSC codes.

We perform experiments on the uni-labelled dataset so as to demonstrate the value of the classifiers when applied to a simpler problem setting. In particular, we attempt to remove any labelling inconsistencies caused by the "Author Labelling Problem", by restricting our labelled data to documents tagged with exactly one label from the top twenty subject areas. In particular, we avoid instances of potential over-labelling. Since this is a similar set-up to the work in [12], we can also directly compare our results.

The uni-labelled setting is not realistic since mathematical documents in digital libraries are usually tagged with multiple labels from a wide range of subject areas. Hence, we also perform our experiments on the full multi-labelled dataset to demonstrate the value of the classifiers in a real-world setting. Furthermore,

this is a similar set-up to the work in [13] and again, we directly compare our results.

In this thesis, we have sufficiently explored latent topic models in the context of mathematical document classification and present four models. Firstly, *Latent Dirichlet Allocation* (LDA) in Chapter 4: a popular latent topic model for documents. Secondly, *Dual Latent Dirichlet Allocation* (DLDA) in Chapter 5: a novel extension to LDA which assumes observations over a dual vocabulary, in particular, models mathematical documents over word and symbol vocabularies. Thirdly, *Pachinko Allocation* (PA) in Chapter 6: another popular document model which is a hierarchical extension to LDA. Finally, *Dual Pachinko Allocation* (DPA) in Chapter 7: a novel extension to PA which assumes observations over dual vocabularies.

We directly address our claim to be tested and demonstrate that accurate multi-label classification requires the inclusion of symbol data. In particular, we see that classification via Dual Pachinko Allocation yields the best performance, and furthermore outperforms our single vocabulary models as well as the existing work proposed in [13].

8.1 Comparison of Models

In this section, we collect all the experimental results for each of the document models we present in this thesis and compare. We first compare the overall classification performance via each document model by comparing the highest Labelling F-Score observed. Recall, the Labelling F-Score measures per-document classification performance and accounts for partial matches. Table 8.1 below shows the highest Labelling F-Score achieved by the classifiers in each document model.

	Multi-Labelled	Uni-Labelled
LDA	65.14%	87.35%
DLDA	60.39%	82.33%
PA	59.58%	80.00%
DPA	70.88%	80.82%

Table 8.1: Highest observed classification performance via each model

Firstly, looking at the multi-labelled dataset, we see that classification via the dual vocabulary DPA model performs best. Furthermore, in Chapter 7 we report the highest achieved micro-averaged F-Score of 69.99% which exceeds the highest value reported in [13] of 67.3%. We conclude that considering mathematical documents covering multiple subject areas, the symbol data coupled with the hierarchical topic/super-topic structure yields better detection of multiple subject areas of a document when compared with existing methods.

On the other hand, looking at the uni-labelled results, we see that classification via single vocabulary LDA performs best. Here we conclude that when a considering mathematical documents which we assume to cover exactly one of the top twenty subject areas, the word co-occurrences are descriptive enough to discriminate between these subject areas. When introducing mathematical notational information and/or a hierarchical topic/super-topic structure, we see a decline in performance; these new models are potentially overfitting to the data.

In the following sections, we investigate whether the adapted models are introducing unnecessary complexity for this simple scenario yielding confusion between subject areas, or that these models "over-classify" the potentially under-labelled documents. In particular, we compare the per-class true positive, and false positive rates and evaluate the impact on overall performance.

True Positive Rates

We first compare the per-class true positive rates of each of the models: for each class i, we evaluate the percentage of condition positives of class i which are correctly predicted to belong to class i.

When considering the uni-labelled dataset, the true positive rate is somewhat robust to the possibility of under-labelled documents appearing in the training data since the per-class true positive rates are invariant to changes in false positives within classification instances yielding true positives. On the other hand, if a classifier predicts multiple appropriate classes but does not successfully predict the "true" class, then this will indeed have a negative effect on the true positive rates here.

Figure 8-1 shows the distributions of the per-class true positive rates of each model.

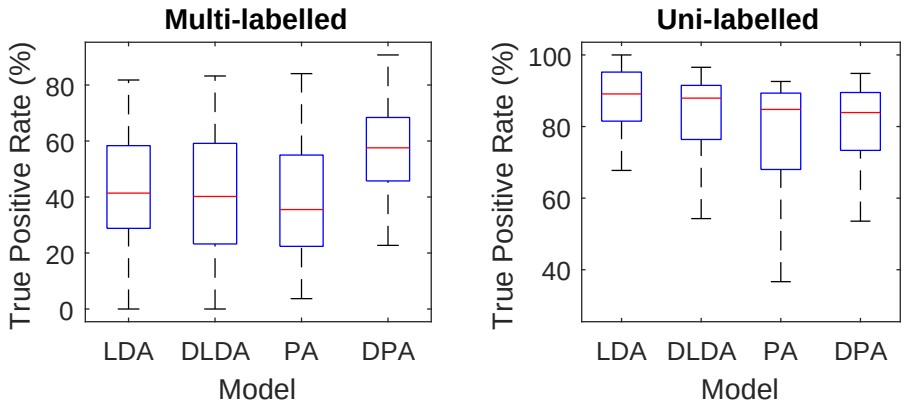

Figure 8-1: Per-class true positive rates of each model

Firstly, looking at the multi-labelled scenario, we see that classification via DPA yields the highest performance. We observe that the lower end of the DPA true positive rates is around 20%, where under LDA, DLDA, and PA, we observe true positive rates which are close to zero. This observation suggests that DPA improves upon the previous models by improving detection on the worst performing categories. Furthermore, we also observe that classification via LDA, DLDA and PA yield similar distributions of true positive rates, whereas classification via DPA yields consistently higher rates and suggests that Dual Pachinko Allocation yields the most discriminative feature set compared to the other models for multi-labelled classification.

We now look at the uni-labelled scenario. We see that compared with LDA, each of the other modes introduce both a decrease and a negative skew to the true positive rates. When considering the decline of true positive rates as an increase in false negative rates, it is evident that the new models may be overfitting to the data which yields a decrease in performance over the simpler model. Furthermore, the skew suggests that feature selection imposed by these new models introduces no extra discriminative properties, but also a feature set which is vaguer across the worst performing subject areas.

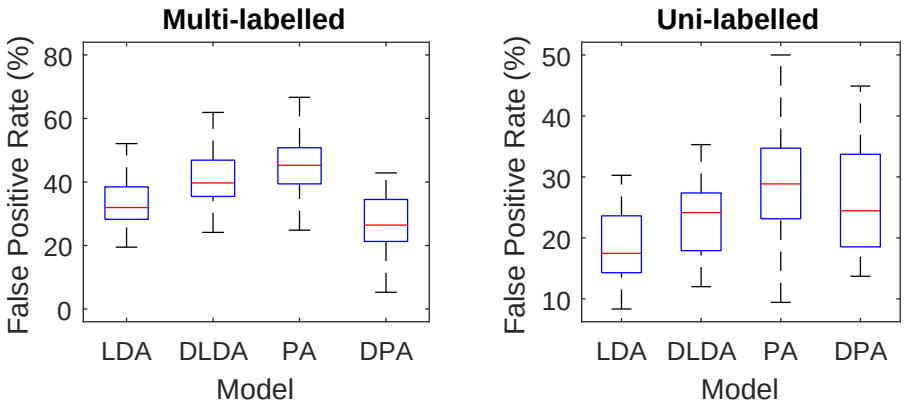

Figure 8-2: Per-class false positives of each model

False Positive Rates

Finally, we look at the per-class false positive rates. That is, for each class i, we evaluate the percentage of condition negatives of class i which are false positives of class i. Figure 8-2 shows the distributions of the false positive rates of each model.

Firstly, looking at the multi-labelled scenarios, we again see that classification via DPA yields the highest performance. On both labelling scenarios, we see that classification via DLDA and PA yield higher false positive rates when compared to classification via LDA. This observation suggests that DLDA and PA yields a feature set that does not correspond well to mathematical subject areas.

Secondly, looking at the uni-labelled scenario, we see that classification via LDA yields the highest performance. We see that each of the other models yield higher rates of false positives. This observation again suggests that the feature selection imposed by the new models reduce the level of discrimination between subject areas.

To summarise, multi-labelled classification via DPA yields strong improves in all aspects; we see both higher true positive rates and lower false positive rates. We conclude that in a realistic problem setting; the structure of mathematical documents strongly adhere to a framework of mixtures of terminologies (topics) and mixtures of notations (symbol topics). Furthermore, amongst the mathematical subject areas according to top-level MSC codes, these mixtures are consistently similar.

We do however see a decrease in classification performance via DPA in the uni-labelled scenario. Given that it is more realistic that documents cover multiple subject areas, it is understandable that powerful models such as DPA yield higher false positive rates in the uni-labelled scenario, however, we also observe lower true positive rates and conclude that classification via LDA within the uni-labelled scenario is most suitable. We conclude that mathematical documents labelled with exactly one of the top twenty top-level MSC subject areas exhibit consistent mixtures of topics between subject areas.

8.2 Future Work

In this section, we propose future directions of research to be considered to continue and improve the work presented in this thesis. In particular, we highlight further research on latent topic models which we could adapt and refine to address the problem field of mathematical document classification, and similarly, how the research presented in this thesis may be adapted to address alternative problem domains.

8.2.1 Hierarchical Dirichlet Processes

We have spent a considerable amount of time on running experiments on different combinations of the numbers of latent topics and super-topics to identify the optimal dimensional settings. An improvement here would be to have these values to be automatically determined in the training process. We could achieve this using Hierarchical Dirichlet Processes [44].

The *Hierarchical Dirichlet Process* (HDP) model is a generalisation of LDA which assumes that observations can be characterised by a mixture of latent topics, but relaxes the assumption the value of K, the number of latent topics, is known and fixed. In [44], the authors present various methods of inference, including Gibbs sampling. Progress has been made in producing other inference methods. In particular, [45] presents a Variational Bayes algorithm and furthermore, [46] provides Online Variational Bayes for Hierarchical Dirichlet Processes.

The papers mentioned here use the HDP model as a purely unsupervised model; HPD is purely used as a tool for topic discovery and document classifica-

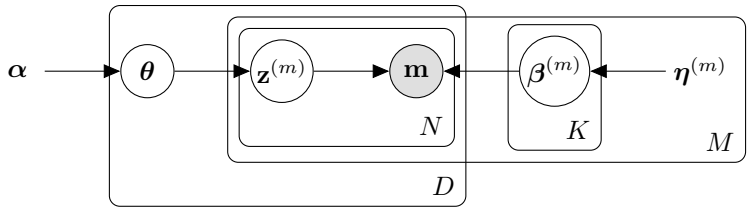

Figure 8-3: Graphical model representation of multi-LDA

tion via HDP is not directly explored.

Finally, HDP provides inspiration for constructing a possible *Hierarchical Pachinko Process* model: a generalisation of Pachinko Allocation which relaxes the assumptions that both the number of topics K and super-topics S are known and fixed.

8.2.2 Multi-vocabulary Approaches

In this thesis, we have demonstrated the value of dual-vocabulary approaches to mathematical document classification. Furthermore, dual vocabulary approaches are used in the contexts of automatic image annotation [47], and author entity resolution [42].

An interesting further generalisation of the single vocabulary models would be the introduction of multi-vocabulary models, which model observations over $M \geq 2$ vocabularies. In particular, Figure 8-3 outlines a graphical model of multi-vocabulary generalisation of Latent Dirichlet Allocation over M fixed vocabularies. Figure 8-4 outlines the same multi-vocabulary generalisation of Pachinko Allocation.

We have shown that when adapting the single vocabulary models to produce the dual vocabulary models, we introduce only minimal changes to the mathematical structure of the model. In particular, the changes are simply additions rather than explicit changes to the existing models, e.g. we may reuse components of the existing models and simply add steps to the existing algorithms. It is likely the case that we may generalise for multi-vocabulary models in the same way, but further research applied to appropriate problem domains is required.

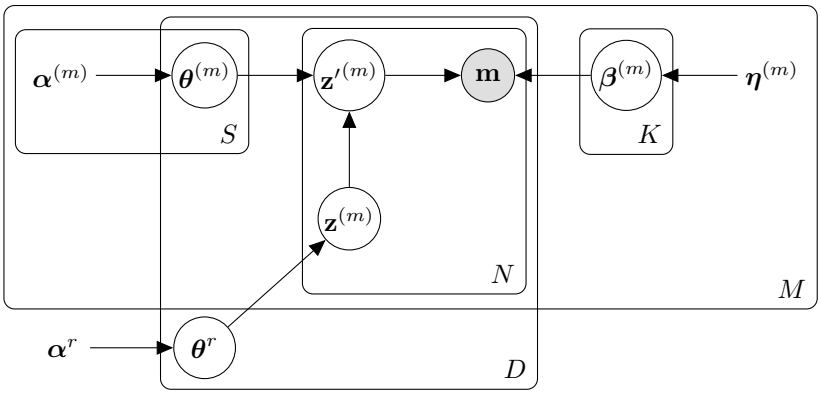

Figure 8-4: Graphical model representation of multi-PA

8.2.3 Mathematical Structure

In this thesis, we follow the bag-of-words approach: we consider documents as collections of unordered words. Furthermore, we consider mathematical documents as collections of unordered words and unstructured symbols. The NTCIR dataset which we use for our training data includes mathematical structural information via the MathML representations which we discard when using the latent topic models outlined in this thesis. We now discuss the possibility of using document structure as well as the mathematical structure in the context of mathematical document modelling.

Firstly, in the computer vision community, latent topic models which incorporate spatial structure information have proven to be useful for image classification. In particular, [48] shows that object detection and classification in photographs is significantly improved when including spatial structure information since nearby visual features are likely to belong to the same visual topic.

In the context of mathematical document modelling, we might assume the same about nearby symbols. In particular, at the document level, symbols which appear in nearby pieces of mathematics (for example, appearing in the same equation or paragraph) are perhaps more likely to belong to the same notation than symbols appearing elsewhere.

Secondly, since MathML is essentially a graph based representation, we may look at graph-based approaches to classification. Graph-based matching is another popular strategy for image classification. For example, in [49], the authors

present the possibility of using frequently occurring sub-graphs of visual features for image classification. In the context of mathematical document classification, this could extend to detecting the commonly occurring sub-expressions to use for classification. For example, the sub-expression "$\frac{\partial}{\partial x}$" is likely to provide more discriminative information than the symbols "∂" and "x" alone. These approaches link well with sub-expression matching and searching [50].

Chapter 9

Concluding Remarks

In this chapter, we give our concluding remarks of the research presented in this thesis. We begin by outlining the contributions of this thesis in the scope of the machine learning and mathematical knowledge management communities. We finish by summarising the outcomes of the experimental results in this thesis and draw conclusions in relation to our research goals.

9.1 Contributions

In this section, we outline the contributions of this thesis to the machine learning and mathematical knowledge management communities. Our contributions in this thesis fall into two categories: firstly, contributions to latent topic modelling, and secondly, contributions to mathematical document classification.

9.1.1 Latent Topic Modelling

We first summarise our contributions to document modelling: we outline our contributions which are not specific to the problem domain of the modelling of mathematical documents, we outline our contributions which can be applied to any scenario which involves modelling collections of discrete data.

Online Pachinko Allocation

In Chapter 6, we formally outline Variational Bayes for Pachinko Allocation, and furthermore, introduce a novel extension to give an Online Variational Bayes

algorithm for Pachinko Allocation. Online Variational Bayes for Pachinko Allocations provides computationally efficient inference over large collections of data. In particular, this method requires only a single pass of the data and very useful for realistic scenarios where data arrives in a stream; this is in contrast to typical Gibbs sampling methods which have high memory requirements and require many iterations of the data.

Dual Pachinko Allocation

In Chapter 7, we introduce Dual Pachinko Allocation: a novel generalisation of Pachinko Allocation which assumes observations over a dual vocabulary. Furthermore, we present Online Variational Bayes for Dual Pachinko Allocation: a fast and efficient inference algorithm which requires only a single pass of the training data. The DPA model relaxes the constraint imposed by the DLDA model that there is a one-to-one corresponded between the topics of each feature space. We demonstrate the performance of classification via DPA by classifying the subject area of mathematical documents and observe significant gains classification performance when compared to current approaches.

By design, the use of the DPA model is not restricted to mathematical document modelling. In particular, DPA can be used to model arbitrary collections of discrete data where the observations span two distinct vocabularies. For example, similar to [42] which models documents as collections of words and author names, or [47] which models annotated images as collections of words and visual features.

9.1.2 Mathematical Document Classification

Finally, we outline our contributions to mathematical knowledge management, in particular, contributions to mathematical document classification.

Mathematical Document Classification via LDA

In Chapter 4, we present mathematical document classification via Latent Dirichlet Allocation. Classification via LDA is not a novel concept, but until now has not been explored in the context of mathematical document classification. We show that mathematical document classification via LDA performs extremely well

on the simple uni-labelled scenario and is directly comparable in performance to the work presented in [12].

We have also shown that not only LDA performs well as a base for mathematical document modelling, we also show that LDA extends to classify mathematical documents better via DPA which we discuss in the next section. The success of DPA demonstrates the value of LDA as a base model opens doors to investigating other LDA variants which we discuss in Section 8.2.

Mathematical Document Classification via DPA

In Chapter 7, we present the novel latent topic model Dual Pachinko Allocation where classification via Dual Pachinko Allocation directly addresses the problem of mathematical document classification. Furthermore, classification via DPA allows us to address this issue in a principled way using strong statistical and Bayesian foundations.

The authors of [13] allude to the possibility of using symbol data to improve classification performance but do not explore this in detail. Furthermore, in [13], the proposed idea is to treat symbols like words and continue to use a single vocabulary.

The novelty of Dual Pachinko Allocation is that DPA yields a feature set of separate word topics and symbol topics, where these topics are formed conditionally independently from one another given the documents' super-topic mixtures. The assumptions made under DPA allow us to address our problem in a principled way. In particular, we base our model on the statistical structure of mathematical documents opposed to a purely discriminative approach.

The experimental results in Chapter 7 show that mathematical document classification via Dual Pachinko Allocation yields higher performance in a realistic problem setting than the work presented in [13].

9.2 Conclusion

To conclude, we first refer to our research goals:

1. Explore latent topic models in the context of mathematical document modelling. In particular, evaluate classification performance via Latent Dirich-

let Allocation and Pachinko Allocation and compare to the prior art of mathematical document classification.

2. Explore the use of symbol data applied to mathematical document classification. In particular, extend the Latent Dirichlet Allocation and Pachinko Allocation models to operate of dual vocabularies and again evaluate classification performance and compare with the prior art.

3. Explore Online Variational Bayes for higher level models. In particular, develop fast and efficient inference techniques for our dual vocabulary models and Pachinko Allocation which require only a single pass of the data.

In this thesis, we have directly addressed our three research goals. In particular, we have explored mathematical document classification via the existing latent topic models Latent Dirichlet Allocation and Pachinko Allocation. Furthermore, we have introduced and explored novel extensions to these models so as to include mathematical notational data also, namely Dual Latent Dirichlet Allocation and Dual Pachinko Allocation. Finally, we have presented novel Online Variational Bayes algorithms fo DLDA, PA and DPA which allow us to perform significant numbers of experiments quickly. We have performed a wide range of experiments on each of these document models and evaluated classification performance have directly compared our results to the existing work in [12] and [13] and discover the strongest classification performance via LDA and DPA.

9.2.1 Single Vocabulary Approaches

We now summarise the results of our experiments for each model in relation to our research aims. We first discuss the single vocabulary models introduced in this thesis and draw conclusions in the context mathematical document modelling.

Latent Dirichlet Allocation

Latent Dirichlet Allocation proves to be a very useful tool for mathematical document classification: classification via LDA performs very well on both datasets and is comparable in performance to [12] and [13]. Given the success of classification via LDA on the uni-labelled scenario coupled with the decline of performance

of the other models, we conclude that the simple mathematical document structure assumed by LDA is the best fit. In particular, documents that belong to exactly one of the top twenty top-level MSC subject areas have similar mixtures of terminology (topics) but do not necessarily share the same mixtures of symbol notations.

Finally, LDA provides interesting tools and techniques which we use and adapt for the remaining models. In particular, Batch and Online Variational Bayes methods, which allows for the computationally efficient inference of the extended models presented in this thesis.

Pachinko Allocation

Classification via Pachinko Allocation, a stronger variant of LDA, unfortunately, yields the poorest classification performance amongst the models presented in this thesis. In this context of mathematical document classification, this suggests that an elaborate hierarchical topic structure imposed by Pachinko Allocation overfits the data and is unnecessary for detecting the subject areas according to the top level MSC codes. In particular, this observation supports the goodness of fit of the LDA model to the data: the mathematical documents exhibit robust and consistent mixtures of terminologies (topics) between subject areas.

On the other hand, Pachinko Allocation proves to be a good stepping stone to Dual Pachinko Allocation which we summarise later. In particular, Online Variational Bayes for LDA extends nicely into PA and in turn DPA which yields significantly strong classification performance.

9.2.2 Dual Vocabulary Approaches

We now discuss classification via our dual vocabulary models: Dual Latent Dirichlet Allocation and Dual Pachinko Allocation. These models both assume observations over dual vocabularies which detect correlations between words and symbols at different levels. We now summarise the performance of classification via these models, and again, draw conclusions in the context mathematical document modelling.

Dual Latent Dirichlet Allocation

Firstly, we discuss classification via Dual Latent Dirichlet Allocation. DLDA is our extension to the existing LDA model which further includes mathematical notational data to the document modelling set-up. The main characteristic of DLDA is that the each word and symbol is attributed to a particular topic, where topics are characterised by distributions of words and symbols, and more importantly, there is a one-to-one correspondence between word topics and symbol topics.

We discover, that compared with single vocabulary LDA model, that these extra assumptions impose a decrease in classification performance. This observation may suggest that there is not a one-to-one correspondence between word topics and symbols topics. We investigate this further when considering classification via Dual Pachinko Allocation which we summarise in the next section.

This observation follows intuition. For example, consider the subject area of "Probability theory and stochastic processes" which would understandably have robust and consistent notational conventions over many documents. However, documents which simply employ probabilistic methods (for example, to prove a hypothesis) are likely to share the same notation almost exactly, yet the collections of words are likely to be vastly different. This situation and for other similar subjective areas are liable to be responsible for a poorer fit of word topics since they are forced to become broader.

Dual Pachinko Allocation

Finally, we discuss classification via Dual Pachinko Allocation. We discover that mathematical document classification via DPA proves to be most successful for multi-labelled classification. In particular, multi-labelled classification performance via DPA is higher than the other models in this thesis and the work presented in [13]. Furthermore, we see the strongest gains in performance on the previously lowest-performing categories. Classification via DPA outperforms classification via both the single vocabulary LDA and PA models, which directly supports our claim that accurate mathematical document classification requires symbol notation.

When compared with DLDA, we relax the constraint that there is a one-to-

one correspondence between word and symbol topics. Under DPA, there is a conditional independence of the per-document word and symbol topic mixtures given the per-document super-topic mixtures. Since we see an increase in classification performance via DPA over DLDA, we conclude that there are indeed notational conventions with strong discriminative properties. Furthermore, this supports the claim the there is not a one-to-one correspondence between word topics and symbol topics.

We recall comment regarding subject areas which may simply "employ" mathematical concepts from other subject areas. In this scenario, the DPA model is more appropriate; it allows the approximated word-topics to remain unaffected by the symbols. Therefore, we may represent a document as a mixture of much more refined word topics and symbol topics, as opposed to a mixture of broader topics which are essentially combined word and symbol topics.

9.3 Final Remarks

To summarise, in this thesis, we have explored in depth mathematical document classification via latent topic models. In particular, we focus on the claim that accurate mathematical document classification requires the inclusion of symbol data. We have primarily used semi-supervised approaches where we utilise a partially labelled corpus of mathematical documents for training. We investigate existing latent topic models and develop interesting extensions to operate on separate word and symbol vocabularies. Furthermore, we develop fast and efficient inference methods for parameter estimation.

We conclude that the mathematical documents belonging to exactly one of the most common subject areas exhibit strong similarities of the mixtures of terminologies used, where these mixtures strongly correlate to the mathematical subject areas. Furthermore, we conclude that mathematical documents which cover multiple subject areas also exhibit strong similarities of the mixtures of notational conventions, where the various combinations of word terminology and notational conventions strongly correlate with mathematical subject areas.

Finally, the latent topic models presented in this thesis are not restricted to the domain of mathematical document classification. These models generalise to apply any classification scenario where observations can be assumed as collections

of discrete data spanning dual feature spaces, and furthermore, naturally extends to future directions of research in these respective fields.

List of Figures

List of Tables

Bibliography

[1] S. M. Watt, "Mathematical document classification via symbol frequency analysis," in *Intelligent Computer Mathematics: Proc. AISC/Calculemus/MKM 2008*, vol. 5144 of *Lecture Notes in Computer Science*, pp. 29–40, Springer, 2008.

[2] J. Borwein, E. Rocha, and J. Rodrigues, *Communicating Mathematics in the Digital Era*. CRC Press, 2008.

[3] F.-F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, (Washington, DC, USA), pp. 524–531, IEEE Computer Society, 2005.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[5] W. Li and A. McCallum, "Pachinko allocation: DAG-structured mixture models of topic correlations," in *Proceedings of the 23rd international conference on Machine learning*, ICML '06, (New York, NY, USA), pp. 577–584, ACM, 2006.

[6] M. D. Hoffman, D. M. Blei, and F. R. Bach, "Online learning for latent Dirichlet allocation.," in *NIPS* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds.), pp. 856–864, Curran Associates, Inc., 2010.

[7] F. Cajori, *A History of Mathematical Notations: Vol. II.* A History of Mathematical Notations, Cosimo Classics, 2013.

[8] J. Davenport, "Nauseating notation." `http://staff.bath.ac.uk/masjhd/Notation.pdf`, 2013.

[9] R. Moore, "Ongoing efforts to generate "tagged PDF" using pdfTEX," *Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009*, pp. 125–131, 2009.

[10] World-Wide Web Consortium, "Mathematical markup language (MathML) version 3.0 2nd edition: W3C recommendation 10 April 2014." `http://www.w3.org/TR/2014/REC-MathML3-20140410/`, 2014.

[11] S. Lawrence, "Free online availability substantially increases a paper's impact," *Nature*, vol. 411, no. 6837, 2001.

[12] R. Rehůřek and P. Sojka, "Automated classification and categorization of mathematical knowledge," in *Intelligent Computer Mathematics* (S. Autexier, J. Campbell, J. Rubio, V. Sorge, M. Suzuki, and F. Wiedijk, eds.), vol. 5144 of *Lecture Notes in Computer Science*, pp. 543–557, Springer Berlin Heidelberg, 2008.

[13] S. Barthel, S. Tönnies, and W.-T. Balke, "Large-scale experiments for mathematical document classification," in *Digital Libraries: Social Media and Community Networks*, pp. 83–92, Springer, 2013.

[14] H. Wang, M. Huang, and X. Zhu, "A generative probabilistic model for multi-label classification," in *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pp. 628–637, IEEE, 2008.

[15] J. Lasserre, C. Bishop, J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, "Generative or discriminative? getting the best of both worlds," *Bayesian Statistics*, vol. 8, pp. 3–24, 2007.

[16] T. Xiong and V. Cherkassky, "A combined SVM and LDA approach for classification," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 3, pp. 1455–1459 vol. 3, July 2005.

[17] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[18] G. Heinrich and M. Goesele, "Variational Bayes for generic topic models," in *KI 2009: Advances in Artificial Intelligence* (B. Mertsching, M. Hund, and Z. Aziz, eds.), vol. 5803 of *Lecture Notes in Computer Science*, pp. 161–168, Springer Berlin Heidelberg, 2009.

[19] "2010 Mathematics Subject Classification." `http://www.ams.org/mathscinet/msc/pdfs/classifications2010.pdf`.

[20] "NTCIR-11 Math-2 document set." `http://ntcir-math.nii.ac.jp/data/`.

[21] T. Bray, F. Yergeau, E. Maler, J. Paoli, and M. Sperberg-McQueen, "Extensible markup language (XML) 1.0 (fifth edition)," W3C recommendation, W3C, Nov. 2008. http://www.w3.org/TR/2008/REC-xml-20081126/.

[22] "Mathematical reviews." `http://www.ams.org/mr-database`.

[23] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, pp. 1–47, Mar. 2002.

[24] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[25] M. F. Porter, "Snowball: A language for stemming algorithms." `http://snowball.tartarus.org/texts/introduction.html`, 2001.

[26] Q. Wei and R. L. Dunbrack Jr, "The role of balanced training and testing data sets for binary classifiers in bioinformatics," *PloS one*, vol. 8, no. 7, p. e67863, 2013.

[27] "Balena, University of Bath high-performance computing facility." `https://wiki.bath.ac.uk/display/BalenaHPC/System+Architecture`.

[28] T. Joachims, "Advances in kernel methods," ch. Making Large-scale Support Vector Machine Learning Practical, pp. 169–184, Cambridge, MA, USA: MIT Press, 1999.

[29] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[30] S. S. Wilks, *Mathematical statistics*. A Wiley publication in mathematical statistics, New York: John Wiley, cop. 1962, 1963. An early version of some of the material was issued in 1943.

[31] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[32] J. M. Dickey, "Multiple hypergeometric functions: Probabilistic interpretations and statistical uses," *Journal of the American Statistical Association*, vol. 78, no. 383, pp. 628–637, 1983.

[33] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[34] H. Attias, "A variational Bayesian framework for graphical models," in *In Advances in Neural Information Processing Systems 12*, pp. 209–215, MIT Press, 2000.

[35] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[36] T. P. Minka, "Estimating a Dirichlet distribution," tech. rep., 2000.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[38] K. Krstovski, D. A. Smith, H. M. Wallach, and A. McGregor, "Efficient nearest-neighbor search in the probability simplex," in *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, (New York, NY, USA), pp. 22:101–22:108, ACM, 2013.

[39] W. Yang, L. Xu, X. Chen, F. Zheng, and Y. Liu, "Chi-squared distance metric learning for histogram data," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[40] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda, "Maximal margin labeling for multi-topic text categorization," in *Advances in neural information processing systems*, pp. 649–656, 2004.

[41] D. M. Blei and M. I. Jordan, "Modeling annotated data," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, (New York, NY, USA), pp. 127–134, ACM, 2003.

[42] L. Shu, B. Long, and W. Meng, "A latent topic model for complete entity resolution," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pp. 880–891, IEEE, 2009.

[43] W. Li and A. McCallum, "Pachinko allocation: Scalable mixture models of topic correlations," Citeseer, 2008.

[44] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, 2004.

[45] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.

[46] M. Bryant and E. B. Sudderth, "Truly nonparametric online variational inference for hierarchical Dirichlet processes," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 2699–2707, Curran Associates, Inc., 2012.

[47] W. Chong, D. Blei, and F.-F. Li, "Simultaneous image classification and annotation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1903–1910, IEEE, 2009.

[48] X. Wang and E. Grimson, "Spatial latent Dirichlet allocation," in *Advances in Neural Information Processing Systems 20* (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), pp. 1577–1584, Curran Associates, Inc., 2008.

[49] N. Acosta-Mendoza, A. Gago-Alonso, and J. E. Medina-Pagola, "Frequent approximate subgraphs as features for graph-based image classification," *Knowledge-Based Systems*, vol. 27, pp. 381–392, 2012.

[50] M. Kohlhase, B. A. Matican, and C.-C. Prodescu, *MathWebSearch 0.5: Scaling an Open Formula Search Engine*, pp. 342–357. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[51] "ArXiv e-Print Archive." `http://www.arxiv.org`.

[52] H. Stamerjohanns and M. Kohlhase, "Transforming the arXiv to XML," in *Intelligent Computer Mathematics* (S. Autexier, J. Campbell, J. Rubio, V. Sorge, M. Suzuki, and F. Wiedijk, eds.), vol. 5144 of *Lecture Notes in Computer Science*, pp. 574–582, Springer Berlin Heidelberg, 2008.

[53] I. Ulusoy and C. M. Bishop, "Generative versus discriminative methods for object recognition," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 258–265, IEEE, 2005.

[54] T. Jebara, *Discriminative, generative and imitative learning*. PhD thesis, Massachusetts Institute of Technology, 2001.

[55] Y. D. Rubinstein and T. Hastie, "Discriminative vs informative learning," in *In Proc. Third Int. Conf. on Knowledge Discover and Data Mining*, pp. 49–53, AAAI Press, 1997.

[56] S. A. Nene and S. K. Nayar, "A simple algorithm for nearest neighbor search in high dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 989–1003, Sept. 1997.

[57] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, pp. 325–327, April 1976.

[58] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*, vol. 31. Springer Science & Business Media, 2013.

[59] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427 – 437, 2009.

Appendix A

Mathematical Knowledge Management

A.1 Mathematical Libraries

arXiv

The arXiv project [51] which started in August 1991, is a highly-automated electronic archive and distribution server for scientific research articles. Authors who submit articles are encouraged to submit LaTeX source code as well as a compiled PDF file. At the time of print, the arXiv had recently accepted it one millionth submission[1].

The ArXMLiv project [52] aims to translate the entire collection of the scientific publications from the arXiv from LaTeX source code into XML format (in particular MathML).

A.2 Mathematics Subject Classification

The Mathematics Subject Classification (MSC) scheme [19] is an article labelling scheme designed to categorise mathematical documents into the relevant areas of interest.

[1] Press release:
https://www.library.cornell.edu/about/news/press-releases/arxiv-hits-1-million-submissions-0

11-XX NUMBER THEORY

- **11-00** General reference works
- **11Axx** Elementary number Theory
  - **11A05** Multiplicative structure; Euclidean algorithm; greatest common divisors
  - **11A07** Congruences; primitive roots; residue systems

62-XX STATISTICS

- **62Hxx** Multivariate analysis
  - **62H15** Hypothesis testing
  - **62H17** Contingency Tables

Figure A-1: Snippet of the MSC scheme.

Each MSC code is a five-character alphanumeric string which follows a hierarchical scheme. An MSC code can be two, three or five characters long depending how specific the classification is. The first two characters are numeric which corresponds to which first level class belongs to the document. The third character is a single Latin character (or possibly "-" for some special cases) which corresponds to which specific area of the first level class. The fourth and fifth characters are numeric which corresponds to a more precise area of the first and second level classes. An example of some MSC codes is given in Figure A-1.

Appendix B

Expanding Expectations

In this section, we provide the expanded forms of the expectations given in the evidence lower bounds of the variational distributions of LDA, DLDA, PA and DPA. The expectations come in one of the following six forms:

1. $\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})]$ where $p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})$ denotes the probability that $\boldsymbol{\theta}_d$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\alpha}$.

2. $\mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})]$ where $q(\boldsymbol{\beta}|\boldsymbol{\lambda})$ denotes the probability that each row of $\boldsymbol{\beta}$ is drawn from a Dirichlet distribution with parameter given by each row of $\boldsymbol{\lambda}$ respectively.

3. $\mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})]$ where $p(\boldsymbol{\beta}|\boldsymbol{\eta})$ denotes the probability that each row of $\boldsymbol{\beta}$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\eta}$.

4. $\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d)]$ where $p(\mathbf{z}_d|\boldsymbol{\theta}_d)$ denotes the probability that each row of $\mathbf{z}_d$ is drawn from a Categorical distribution with parameter $\boldsymbol{\theta}_d$.

5. $\mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)]$ where $q(\mathbf{z}_d|\boldsymbol{\phi}_d)$ denotes the probability that each row of $\mathbf{z}_d$ is drawn from a Categorical distribution with parameter given by each row of $\boldsymbol{\phi}_d$ respectively.

6. $\mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})]$ where $p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})$ denotes the probability that the each row $\mathbf{w}_{dn}$ is drawn from Categorical distributions with parameter given by $\boldsymbol{\beta}_{z_{dn}}$.

B.1 Expanding $\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})]$

This is the case where $p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})$ denotes the probability that $\boldsymbol{\theta}_d$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\alpha}$. We start by expanding $p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})$ using to the probability density function of the Dirichlet distribution to give

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] = \mathbb{E}_q\left[\log \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^{K} {\theta_{dj}}^{(\alpha_j - 1)}\right]$$

where K is the length of the row vector $\boldsymbol{\theta}_d$. Rearranging to give a sum of logs yields

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] = -\log B(\boldsymbol{\alpha}) + \sum_{j=1}^{K} (\alpha_j - 1)\mathbb{E}_q[\log \theta_{dj}|\boldsymbol{\alpha}]$$

Finally, expanding $B(\boldsymbol{\alpha})$ yields the expanded expectation

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})] = \log \Gamma\left(\sum_j \alpha_j\right) + \sum_{j=1}^{K} ((\alpha_j - 1)\mathbb{E}_q[\log \theta_{dj}] - \log \Gamma(\alpha_j))$$

B.2 Expanding $\mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})]$

This is the case where $q(\boldsymbol{\beta}|\boldsymbol{\lambda})$ denotes the probability that each row of $\boldsymbol{\beta}$ is drawn from a Dirichlet distribution with parameter given by each row of $\boldsymbol{\lambda}$ respectively. Since each row $\boldsymbol{\beta}_j$ depends on $\boldsymbol{\lambda}$ through $\boldsymbol{\lambda}_j$ only, we factorise $q(\boldsymbol{\beta}|\boldsymbol{\lambda})$ as a product of probabilities

$$q(\boldsymbol{\beta}|\boldsymbol{\lambda}) = \prod_{j=1}^{K} q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j)$$

where K is the number of rows in $\boldsymbol{\beta}$, and $q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j)$ denotes the probability that $\boldsymbol{\beta}_j$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\lambda}_j$. Rearranging to give a sum of logs yields

$$\mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] = \sum_{j=1}^{K} \mathbb{E}_q\left[\log q(\boldsymbol{\beta}_j|\boldsymbol{\lambda}_j)\right]$$

Finally, we note that the inner expectation is of the required form of the case outlined in the previous section. Using the expanded form of this expectation from Appendix B.1 yields the expanded expectation

$$\mathbb{E}_q[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda})] = \sum_{j=1}^{K}\left(\log\Gamma\left(\sum_{v}\lambda_{jv}\right) + \sum_{v=1}^{V}((\lambda_{jv}-1)\mathbb{E}_q[\log\beta_{jv}] - \log\Gamma(\lambda_{jv}))\right)$$

where V is the number of columns in $\boldsymbol{\beta}$.

B.3 Expanding $\mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})]$

This is the case where $p(\boldsymbol{\beta}|\boldsymbol{\eta})$ denotes the probability that each row of $\boldsymbol{\beta}$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\eta}$. Since the rows of $\boldsymbol{\beta}$ are conditionally independent given $\boldsymbol{\eta}$, we factorise $p(\boldsymbol{\beta}|\boldsymbol{\eta})$ as a product of probabilities

$$p(\boldsymbol{\beta}|\boldsymbol{\eta}) = \prod_{j=1}^{K} p(\boldsymbol{\beta}_j|\boldsymbol{\eta})$$

where K is the number of rows in $\boldsymbol{\beta}$ and $p(\boldsymbol{\beta}_j|\boldsymbol{\eta})$ denotes the probability that $\boldsymbol{\beta}_j$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\eta}$. Rearranging to give a sum of logs yields

$$\mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] = \sum_{j=1}^{K}\mathbb{E}_q\big[\log p(\boldsymbol{\beta}_j|\boldsymbol{\eta})\big]$$

Finally, we note that the inner expectation is of the required form outlined in Appendix B.1. Substituting in the expanded form of this expectation yields the expanded expectation

$$\mathbb{E}_q[\log p(\boldsymbol{\beta}|\boldsymbol{\eta})] = K\left(\log\Gamma\left(\sum_{v}\eta_v\right) - \sum_{v=1}^{V}\log\Gamma(\eta_v)\right) + \sum_{v=1}^{V}(\eta_v-1)\sum_{j=1}^{K}\mathbb{E}_q[\log\beta_{jv}]$$

where V is the number of columns in $\boldsymbol{\beta}$.

B.4 Expanding $\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d)]$

This is the case where $p(\mathbf{z}_d|\boldsymbol{\theta}_d)$ denotes the probability that each row of $\mathbf{z}_d$ is drawn from a Categorical distribution with parameter $\boldsymbol{\theta}_d$. Since the rows of $\mathbf{z}_d$ are conditionally independent given $\boldsymbol{\theta}_d$ we can factorise $p(\mathbf{z}_d|\boldsymbol{\theta}_d)$ as a product of probabilities

$$p(\mathbf{z}_d|\boldsymbol{\theta}_d) = \prod_{n=1}^{N_d} p(\mathbf{z}_{dn}|\boldsymbol{\theta}_d)$$

where N_d is the number of rows in $\mathbf{z}_d$ and $p(\mathbf{z}_{dn}|\boldsymbol{\theta}_d)$ denotes the probability that $\mathbf{z}_{dn}$ is drawn from a Categorical distribution with parameter $\boldsymbol{\theta}_d$. Rearranging to give a sum of logs yields

$$\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d)] = \sum_{n=1}^{N_d} \mathbb{E}_q[\log p(\mathbf{z}_{dn}|\boldsymbol{\theta}_d)]$$

Finally, we rearrange using the probability density function of the Categorical distribution to give the expanded expectation

$$\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\theta}_d)] = \sum_{n=1}^{N_d} \sum_{j=1}^{K} \mathbb{E}_q[z_{dnj} \log \theta_{dj}]$$

where K is the number of columns in $\mathbf{z}_d$.

B.5 Expanding $\mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)]$

This is the case where $q(\mathbf{z}_d|\boldsymbol{\phi}_d)$ denotes the probability that each row of $\mathbf{z}_d$ is drawn from a Categorical distribution with parameter given by each row of $\boldsymbol{\phi}_d$ respectively. Since each of the rows $\mathbf{z}_{dn}$ depends on $\boldsymbol{\phi}_d$ through $\boldsymbol{\phi}_{dn}$ only, we factorise $q(\mathbf{z}_d|\boldsymbol{\phi}_d)$ as a product of probabilities

$$q(\mathbf{z}_d|\boldsymbol{\phi}_d) = \prod_{n=1}^{N_d} q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn})$$

where N_d is the number of rows in $\mathbf{z}_d$ and $q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn})$ denotes the probability that $\mathbf{z}_{dn}$ is drawn from a Categorical distribution with parameter $\boldsymbol{\phi}_{dn}$. Rearranging to give a sum of logs yields

$$\mathbb{E}_q[\log q(\mathbf{z}_d|\boldsymbol{\phi}_d)] = \sum_{n=1}^{N_d} \mathbb{E}_q[\log q(\mathbf{z}_{dn}|\boldsymbol{\phi}_{dn})]$$

Finally, we note that the inner expectation is of the required form outlined in Appendix B.4. Substituting in the expanded form of this expectation yields the expanded expectation

$$\mathbb{E}_q[\log p(\mathbf{z}_d|\boldsymbol{\phi}_d)] = \sum_{n=1}^{N_d} \sum_{j=1}^{K} \mathbb{E}_q[z_{dnj} \log \phi_{dnj}]$$

where K is the number of columns in $\mathbf{z}_d$.

B.6 Expanding $\mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})]$

This is the case where $p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})$ denotes the probability that each of the each row $\mathbf{w}_{dn}$ is drawn from Categorical distributions with parameter given by $\boldsymbol{\beta}_{z_{dn}}$. Since each of the rows $\mathbf{w}_{dn}$ depend on $\mathbf{z}_d$ through $\mathbf{z}_{dn}$ only, we factorise $p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})$ as a product of probabilities

$$p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta}) = \prod_{n=1}^{N_d} p(\mathbf{w}_{dn}|\mathbf{z}_{dn}, \boldsymbol{\beta})$$

where N_d is the number of rows of $\mathbf{w}_d$. Using the that we can rewrite $p(\mathbf{w}_{dn}|\mathbf{z}_{dn})$ as the probability density function of the Categorical distribution characterised by $p(\mathbf{w}_{dn}|\boldsymbol{\beta}_{z_{dn}})$, we rearrange to give

$$\mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})] = \sum_{n=1}^{N_d} \mathbb{E}_q\big[\log p(\mathbf{w}_{dn}|\boldsymbol{\beta}_{z_{dn}})\big]$$

Plugging in the probability density function of the Categorical distribution

and rearranging to give a sum of logs yields

$$\mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})] = \sum_{n=1}^{N_d} \sum_{j=1}^{K} \mathbb{E}_q[z_{dnj} \log \beta_{jw_{dn}}]$$

where K is the number of columns in $\mathbf{z}_d$. Rewriting $\beta_{jw_{dn}} = \prod_{v=1}^{V} \beta_{jv}^{w_{dnv}}$ and rearranging yields the summation

$$\mathbb{E}_q[\log p(\mathbf{w}_d|\mathbf{z}_d, \boldsymbol{\beta})] = \sum_{n=1}^{N_d} \sum_{j=1}^{K} \sum_{v=1}^{V} \mathbb{E}_q[w_{dnv} z_{dnj} \log \beta_{jv}]$$

where V is the number of columns in $\mathbf{w}_d$.

Appendix C

Supervised Classification

In this section, we outline various machine learning and document classification techniques. In particular, the methods used in this thesis and similar research areas.

C.1 Discriminative and Generative Models

The main objective of an object classification problem (in the simplest scenario) is to determine the most likely class label $\hat{\mathbf{c}}$ for some data point $\hat{\mathbf{x}}$, given a set of independent training observations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and corresponding labels $\mathbf{c} = \{\mathbf{c}_1, \ldots, \mathbf{c}_N\}$. This requires evaluating the conditional distribution $p(\hat{\mathbf{c}}|\hat{\mathbf{x}}, \mathbf{X}, \mathbf{c})$. In this section, we outline two approaches of determining this distribution as described in [15] and [53]: the *discriminative* approach, and the *generative* approach.

The *discriminative* approach is to determine a probability mapping from the input data $\mathbf{x}$ and the class labels $\mathbf{c}$ over some model parameters $\boldsymbol{\theta}$. We define the conditional distribution $p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of parameters governing the relationship between the input data and the labels. The likelihood function is then given by

$$L(\boldsymbol{\theta}) = p(\mathbf{c}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta})$$

which can be combined with the prior $p(\boldsymbol{\theta})$ to give the joint distribution $p(\boldsymbol{\theta}, \mathbf{c}|\mathbf{X})$

from which we can obtain the posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{c}) = \frac{p(\boldsymbol{\theta})L(\boldsymbol{\theta})}{p(\mathbf{c}|\mathbf{X})}$$

where $p(\mathbf{c}|\mathbf{X}) = \int p(\boldsymbol{\theta})L(\boldsymbol{\theta})d\boldsymbol{\theta}$. Predictions can then be made by marginalising the predictive distribution with respect to $\boldsymbol{\theta}$ weighted by the posterior distribution

$$p(\hat{\mathbf{c}}|\hat{\mathbf{x}}, \mathbf{X}, \mathbf{c}) = \int p(\hat{\mathbf{c}}|\hat{\mathbf{x}}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{c})d\boldsymbol{\theta}$$

In practice these integrations are rarely tractable so approximation must be used. If there is sufficient training data, a point estimate for $\boldsymbol{\theta}$ can be made by maximising the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{c})$ to give $\hat{\boldsymbol{\theta}}$ and the predictive distribution can be approximated using

$$p(\hat{\mathbf{c}}|\hat{\mathbf{x}}, \mathbf{X}, \mathbf{c}) \approx p(\hat{\mathbf{c}}|\hat{\mathbf{x}}, \hat{\boldsymbol{\theta}})$$

Discriminative approaches make no explicit attempt to model the underlying distributions of the variables and features in a system and are only interested in optimising a mapping from the inputs to the desired outputs [54] [55].

By contrast, the *generative* approach is to model the entire system as being generated from a generator $\boldsymbol{\theta}$ (a set of parameters) and then use Bayesian methods to obtain the relevant conditional distribution.

We first model the joint distribution $p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})$ of data points and labels, conditioned on a set of parameters $\boldsymbol{\theta}$. This can be done by learning the class prior probabilities $p(\mathbf{c}|\boldsymbol{\pi})$ for the classes along with the class-conditional densities $p(\mathbf{x}|\mathbf{c}, \boldsymbol{\lambda})$ separately, so that

$$p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta}) = p(c|\boldsymbol{\pi})p(\mathbf{x}|\mathbf{c}, \boldsymbol{\lambda})$$

where $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\lambda}\}$. Since the data points are assumed to be independent, the joint distribution is given by

$$L_G(\boldsymbol{\theta}) = p(\mathbf{X}, \mathbf{c}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{c}_n|\boldsymbol{\theta})$$

which can be maximised to determine the most probable value of $\boldsymbol{\theta}$. Since $p(\mathbf{X}, \mathbf{c}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{c})p(\mathbf{X}, \mathbf{c})$, this is equivalent to maximising the posterior dis-

tribution $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{c})$. The required posterior probabilities can then obtained by using Bayes' theorem:

$$p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})}{\sum_{\mathbf{c}'} p(\mathbf{x}, \mathbf{c}'|\boldsymbol{\theta})}$$

Versatility is inherent when working in the joint distribution space since we can insert knowledge about the relationships between, for example, variables, independence, and prior distributions. In particular, this includes all variables in the system, unobserved, observed, input or output variables which make generative probability distributions a very flexible modelling tool [54]. Compared with discriminative approaches, generative models typically have the advantage that they can handle missing data or partially labelled data [53].

C.2 Nearest Neighbour Methods

The idea of nearest neighbour classification is as follows: given a training set of N observations $\mathbf{x}_n$, and corresponding binary labels $\mathbf{y}_n$, we classify a new observation $\mathbf{x}^*$ by selecting class of the observation in the training set which is the shortest distance from $\mathbf{x}^*$. This process is the *one-nearest neighbour* approach and generalises further to the *K-nearest neighbours* approach by instead selecting the K nearest data points to a new observation and assign the most common label amongst them. Simple nearest neighbour search algorithms are given in [56].

We may wish to weight the evidence of neighbours close to an unclassified observation more heavily than the instances which are at a greater distance. We may achieve a lower probability of misclassification by using a weighted K nearest neighbour classification [57]. In the experiments in this thesis, we use an inverse-distance weighting. That is, we weight the influence of the training data by the inverse of the distance of each training observation to $\mathbf{x}^*$.

Formally, we define the K-nearest neighbour classification according to [58]: given a training set of N observations $\mathbf{x}_n$, we assign a new observation $\mathbf{x}^*$ the value $g(\mathbf{x}^*)$ given

$$g(\mathbf{x}^*) = \begin{cases} 1 & \text{if } \sum_{n=1}^{N} w_n \delta(y_n, 1) > \sum_{n=1}^{N} w_n I_{[y_n=0]} \\ 0 & \text{otherwise} \end{cases}$$

where w_n denotes the weighting for observation $\mathbf{x}_n$. The decision depends on K and $\mathbf{x}^*$ through the weight vector $\mathbf{w}$, for example, the inverse-distance weighting for K nearest neighbour classification is given by

$$w_n = \begin{cases} d(\mathbf{x}_n, \mathbf{x}^*)^{-1} & \text{if } \mathbf{x}_n \text{ is among the } K \text{ nearest neighbours of } \mathbf{x}^* \\ 0 & \text{otherwise} \end{cases}$$

where d is some distance function.

Distance Metrics

We may use any distance metric in the weights which determine the K nearest neighbours. In our experiments, the observations reside in the probability simplex: the space of vectors whose entries lie between zero and one inclusive and sum to one. The usual Euclidean distance metric is not appropriate here; instead, we choose to use the χ^2 distance given by

$$d(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^{K} \frac{(x_i - y_i)^2}{x_i + y_i}$$

Efficient nearest neighbour search algorithms in the probability simplex are described in [38].

Decision Thresholds

Under the above formulation, given an observation $\mathbf{x}^*$, we assign it the most label common label amongst the K *closest* neighbours (according to a pre-specified weighting and distance metric). In some scenarios, including the work in this thesis, we may wish to assign a positive label if there is a sufficient number of examples which are close, but perhaps not the closest. We address this by refactoring the classification process as a score based system.

We define the score of a positive prediction as follows:

$$\text{score}(\mathbf{x}^*) = \frac{\sum_{n=1}^{N} w_n I_{[y_n=1]}}{\sum_{n=1}^{N} w_n}$$

which takes values in the interval $[0, 1]$. Using this score function, we rewrite the decision function $g(\mathbf{x}^*)$ as

$$g(\mathbf{x}^*) = \begin{cases} 1 & \text{if score}(\mathbf{x}^*) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Instead of predicting a positive class when the score is greater than the fixed value 0.5, we may instead use an arbitrary value $t \in [0, 1]$ which gives our the decision threshold based decision function to

$$g(\mathbf{x}^*) = \begin{cases} 1 & \text{if score}(\mathbf{x}^*) > t \\ 0 & \text{otherwise} \end{cases}$$

Multi-label Classification

We can generalise binary classification (where each observation belongs to exactly one of two classes) to multi-label classification (where each observation may belong to multiple classes simultaneously) by constructing an ensemble of per-class binary classifiers.

Given an ensemble of classifiers $g_1, \ldots, g_C$ which predict whether a document belongs to classes $1, \ldots, C$ respectively. The multi-labelled decision function becomes

$$g(\mathbf{x}^*) = [g_1(\mathbf{x}^*), \ldots, g_C(\mathbf{x}^*)]$$

C.3 Performance Measures

In this section, we outline the performance measures used in this thesis. In particular, we introduce the performance measures for multi-label and multi-class classification described in [59].

Notation Suppose we have a set of D labelled documents across C classes. We let L_d and $\hat{L}_d$ denote the true and predicted labels of the dth document

respectively, where

$$L_{di} = \begin{cases} 1 & \text{if document } d \text{ is condition positive of class } i \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{L}_{di} = \begin{cases} 1 & \text{if document } d \text{ is predicted to belong to class } i \\ 0 & \text{otherwise} \end{cases}$$

C.4 Multi-label Performance Measures

We define the following multi-label performance measures.

Exact Match Ratio The *Exact Match Ratio* [40] measures per-document exact classification performance and is given by

$$\text{ExactMatchRatio} = \frac{1}{D} \sum_{d=1}^{D} \delta(\mathbf{L}_d, \hat{\mathbf{L}}_d)$$

The Exact Match Ratio an all-or-nothing measure, any document which has a single misclassification will count negatively towards this score. In this thesis, the level of labelling between authors is inconsistent, and thus the Exact Match Ratio turns out to be a fairly unhelpful performance measure. For example, an author may not bother to include labels for the minor categories appearing in a document yet a strong classifier may correctly predict a positive label on these minor categories but will contribute negatively to the exact match ratio. These inconsistencies between authors are the basis of the "Author Labelling Problem" which we describe in Section 3.2.

Labelling F-Score The *Labelling F-Score* [40] measures per-document classification performance including partial matches and is given by

$$\text{LabellingF-Score} = \frac{1}{D} \sum_{d=1}^{D} \frac{2 \sum_{i=1}^{C} L_{di} \hat{L}_{di}}{\sum_{i=1}^{C} L_{di} + \hat{L}_{di}}$$

This measure evaluates performance at the document level. In particular, how well a classifier labels an unseen document. This is the most appropriate performance measure for the experiments in this thesis.

C.5 Multi-class Performance Measures

In this section, we present some performance measures for this scenario given in [59]. The multi-class scenario, in contrast to the multi-label scenario, is when the observations take exactly one of the multiple labels.

The micro-averaged performance measures present below provide an insight into the per-class effectiveness of multi-labelled classification and are the performance measures which appear in [12] and [13]. We present these multi-class performance measures, in particular, the micro-averaged F-Score so we can directly compare our results.

We define the performance measures in this section in terms of per-class true positives tp_i, true negatives tn_i, false positives fp_i, and false negatives fn_i which are given by

$$\mathrm{tp}_i = \sum_{d=1}^{D} L_{di}\hat{L}_{di} \qquad \mathrm{fp}_i = \sum_{d=1}^{D}(1 - L_{di})\hat{L}_{di}$$
$$\mathrm{tn}_i = \sum_{d=1}^{D}(1 - L_{di})(1 - \hat{L}_{di}) \qquad \mathrm{fn}_i = \sum_{d=1}^{D} L_{di}(1 - \hat{L}_{di})$$

Average Accuracy The Average Accuracy measures the average per class effectiveness of a classifier and is given by

$$\mathrm{AverageAccuracy} = \frac{1}{C}\sum_{i=1}^{C}\frac{\mathrm{tp}_i + \mathrm{tn}_i}{\mathrm{tp}_i + \mathrm{fn}_i + \mathrm{fp}_i + \mathrm{tn}_i}$$

This measure is fairly useless with large numbers of classes; for example, an ensemble of trivial (and useless) "reject all" classifiers on balanced datasets will have very high Average Accuracy due to the large numbers of true negatives.

Micro-averaged Precision The Micro-averaged Precision measures the average per class agreement of the positive labels returned by the classifiers and is given by

$$\mathrm{Precision}_{\mu} = \frac{\sum_{i=1}^{C}\mathrm{tp}_i}{\sum_{i=1}^{C}\mathrm{tp}_i + \mathrm{fp}_i}$$

A classifier will yield high micro-averaged precision for consistently small numbers of per-class false positives.

Micro-averaged Recall The micro-averaged Recall measures the average per class effectiveness of identifying the correct labels of a document and is given by

$$\text{Recall}_{\mu} = \frac{\sum_{i=1}^{C} \text{tp}_i}{\sum_{i=1}^{C} \text{tp}_i + \text{fn}_i}$$

A classifier will yield high micro-averaged Recall for consistently small numbers of per-class false negatives.

Micro-averaged F-Score The micro-averaged F-Score measures the overall per-class effectiveness of a classifier and is given in terms of the micro-averaged Precision and recall:

$$\text{F-Score}_{\mu} = \frac{2\text{Precision}_{\mu}\text{Recall}_{\mu}}{\text{Precision}_{\mu} + \text{Recall}_{\mu}}$$

A classifier will yield a high micro-averaged F-Score for both high micro-averaged Precision and Recall.

Appendix D

Multi-labelled Confusion

Figures D-1 to D-4 show heat-maps highlighting the per-class confusion of of multi-labelled classification via LDA, DLDA, PA and DPA. We evaluate the confusion between classes by calculating the following three classification/misclassification rates:

1. The true positive rates of each class i: the percentages of condition positives of class i that are true positives.

2. The false negative rates of each class i over each other class j: the percentages condition positives of class i which are both false negatives on class i and false positives on class j.

3. The null classification rates of each class i: the percentage of contrition positives of class i that have been predicted to belong to exactly zero classes.

The entries on the diagonal correspond to the per-class true positive rates, and the ijth entry denotes the false negative rates of class i broken down by false positives over class j. The final column corresponds to the rates of null predictions of each class; it is possible for the classifiers to yield no positive predictions. The intensities of green and red correspond to the strengths of the rates of classification and misclassification respectively, and we sort the rows and columns via the true positive rates on the diagonal.

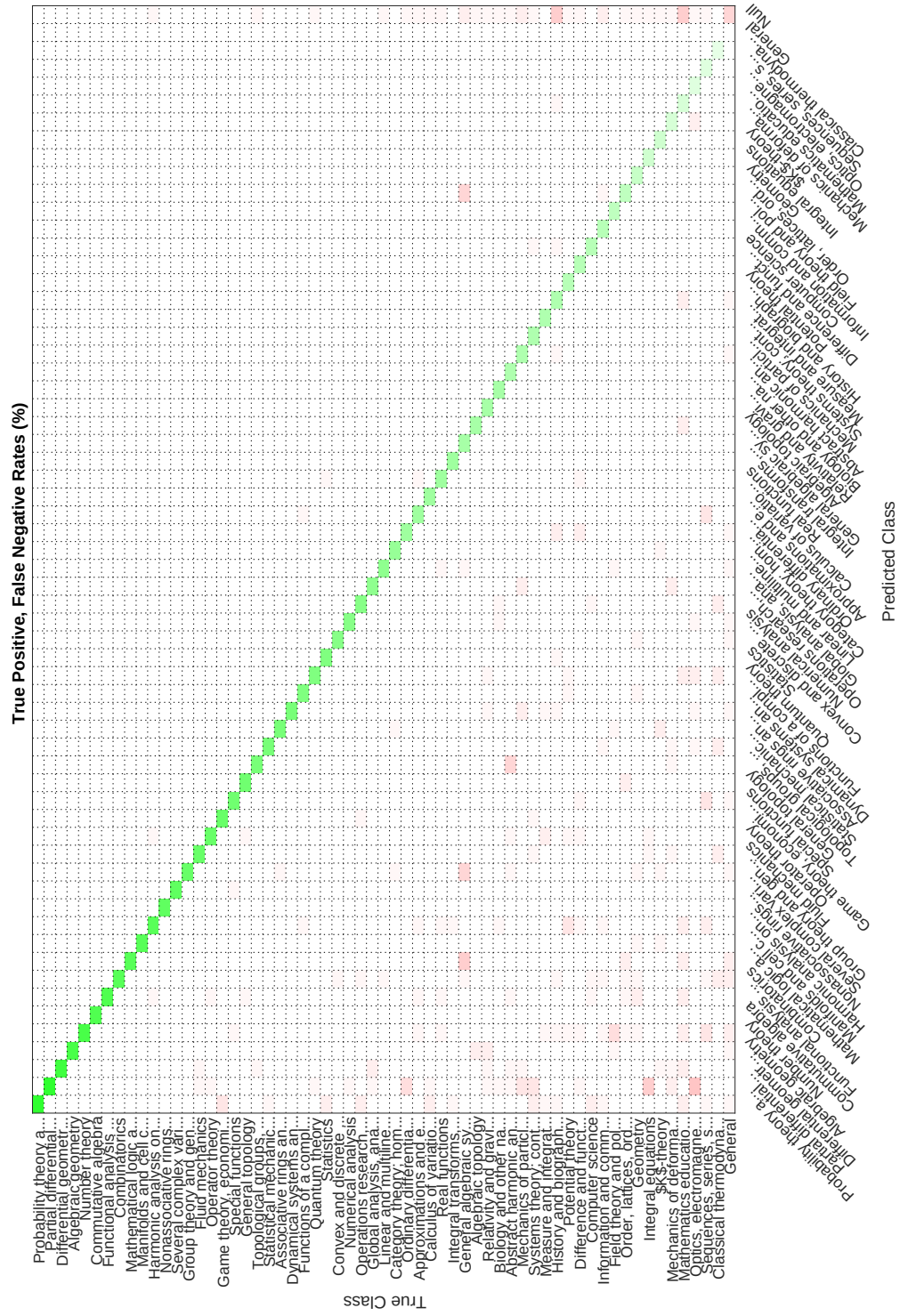

Figure D-1: Per-class confusion of classification via LDA on the multi-labelled dataset - Heat-map

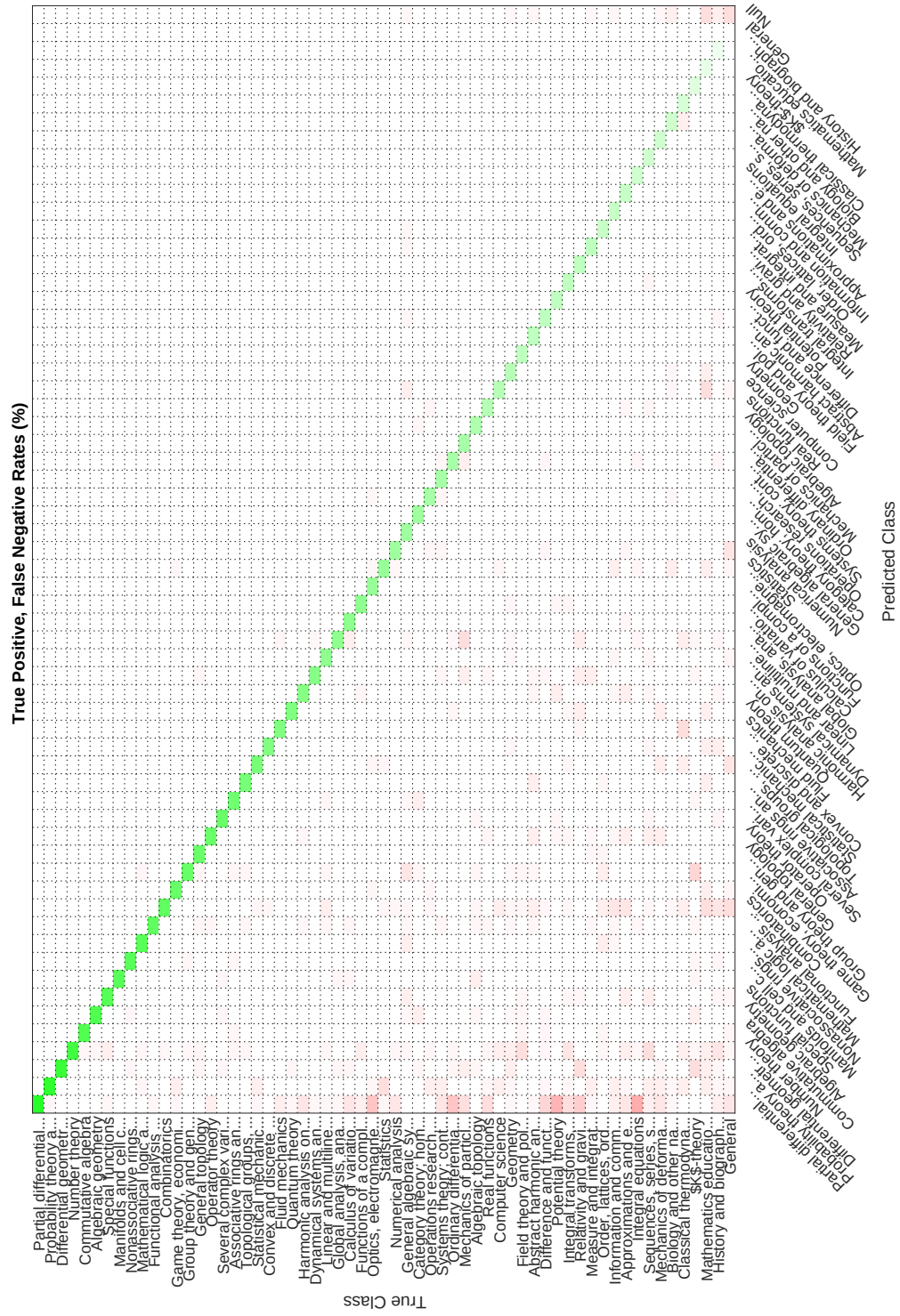

Figure D-2: Per-class confusion of classification via DLDA on the multi-labelled dataset - Heat-map

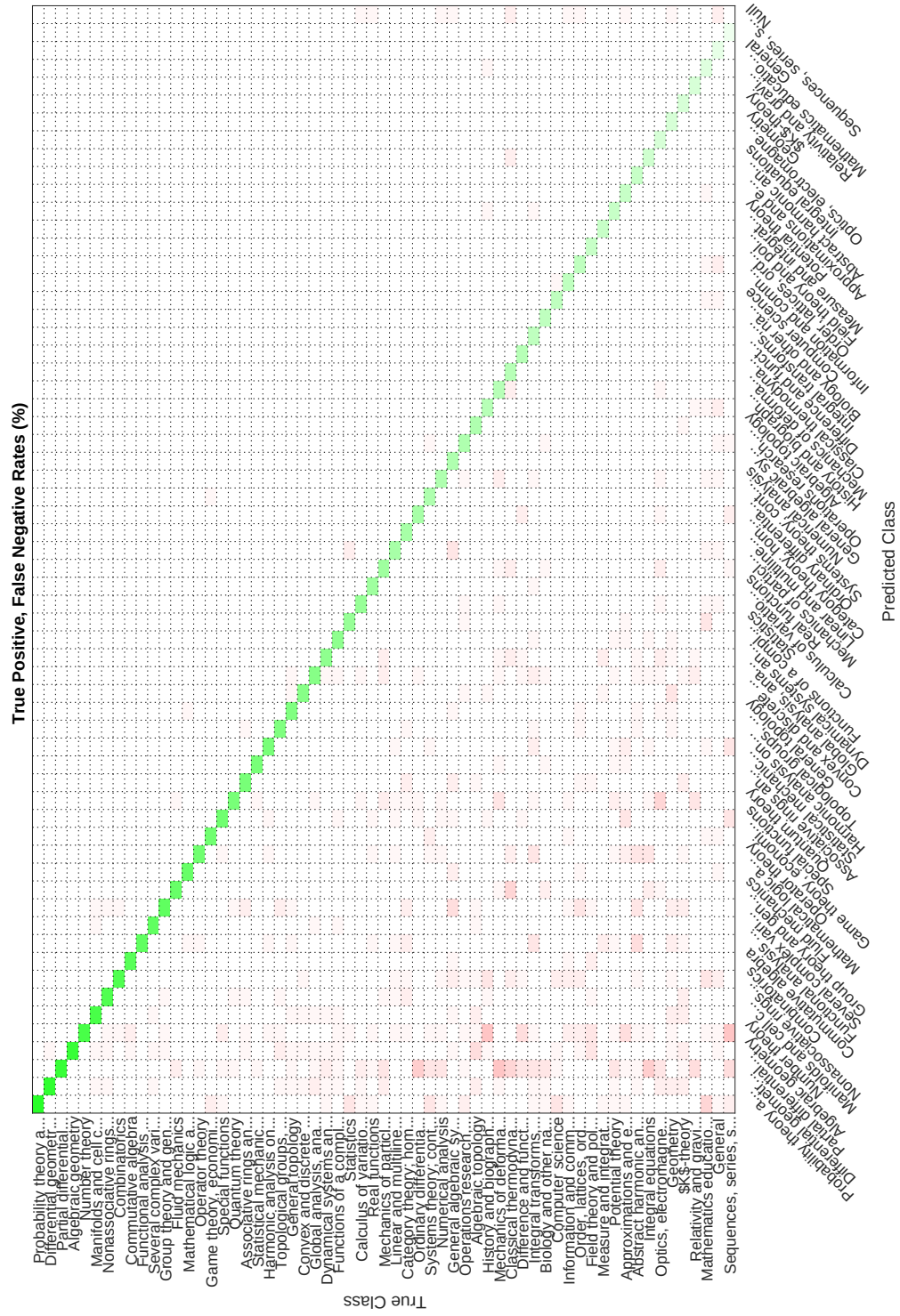

Figure D-3: Per-class confusion of classification via PA on the multi-labelled dataset - Heat-map

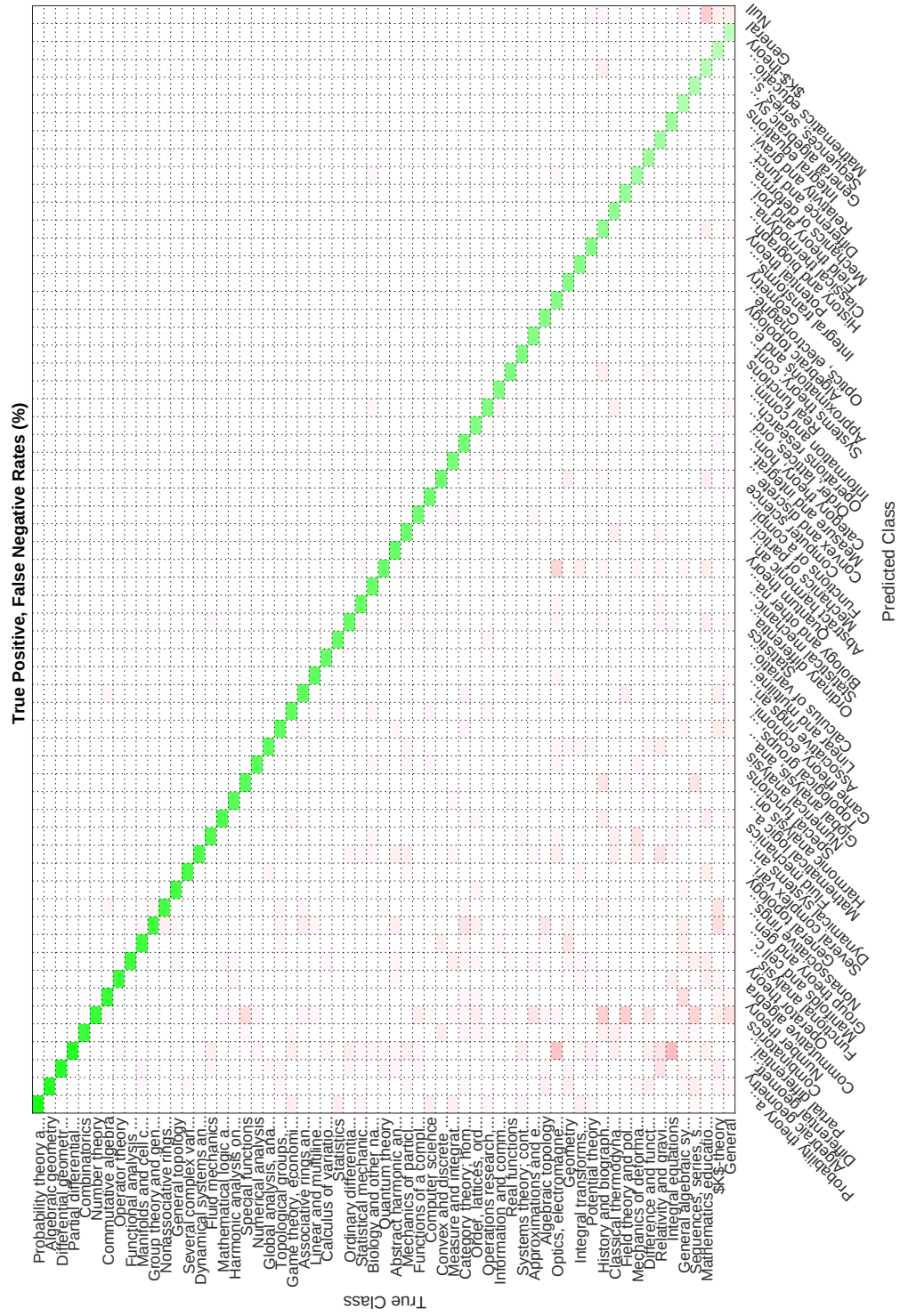

Figure D-4: Per-class confusion of classification via DPA on the multi-labelled dataset - Heat-map

Appendix E

MathML Examples

In this section, we provide example code for MathML presentaion markup and MathML content markup.

E.1 Presentation MathML

Below we provide example code for the MathML *presentation* markup of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$.

```
<mrow>
  <mi>cos</mi>
  <mi>A</mi>
  <mo>=</mo>
  <mn>1</mn>
  <mo>-</mo>
  <mn>2</mn>
  <msup>
    <mi>sin</mi>
    <mn>2</mn>
  </msup>
  <mfrac>
    <mi>A</mi>
    <mn>2</mn>
```

```
    </mfrac>
  <mrow>
```

Listing E.1: Example MathML presentation markup

Figure E-1 below provides a graphical illustration of the above MathML presentation markup.

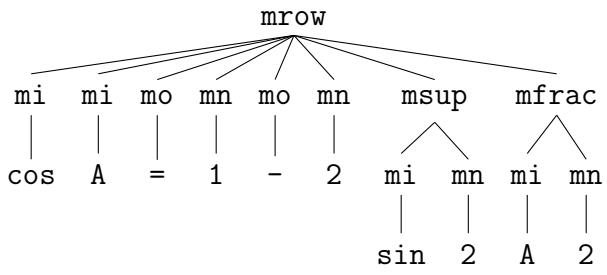

Figure E-1: Graph of the MathML presentation markup of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$

E.2 Content MathML

Below we provide example code for the MathML *content* markup of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$.

```
  <apply>
    <eq/>
    <apply>
      <cos/>
        <ci>A</ci>
    </apply>
    <apply>
      <minus/>
        <cn>1</cn>
        <csymbol>
          <apply>
            <power/>
```

```
            <apply>
              <sin/>
                <csymbol>
                  <apply>
                    <divide/>
                      <ci>A</ci>
                      <cn>2</cn>
                  </apply>
                </csymbol>
            </apply>
             <cn>2</cn>
          </apply>
        </csymbol>
    </apply>
  </apply>
```

Listing E.2: Example MathML content markup

Figure E-2 below provides a graphical illustration of the above MathML content markup.

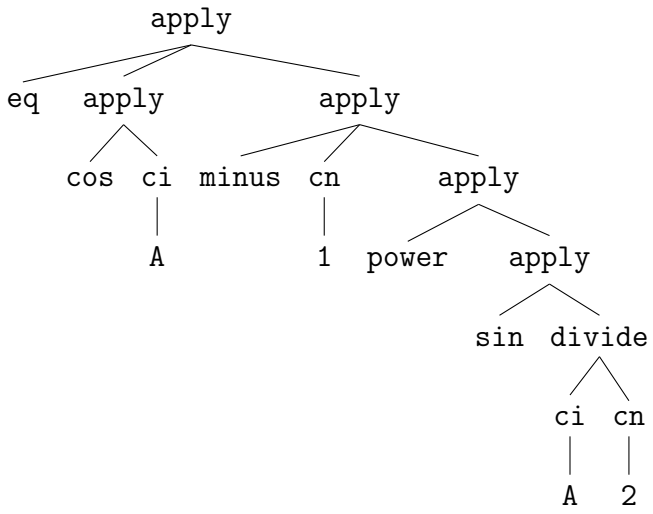

Figure E-2: Graph of the MathML content markup of the expression $\cos A = 1 - 2\sin^2 \frac{A}{2}$